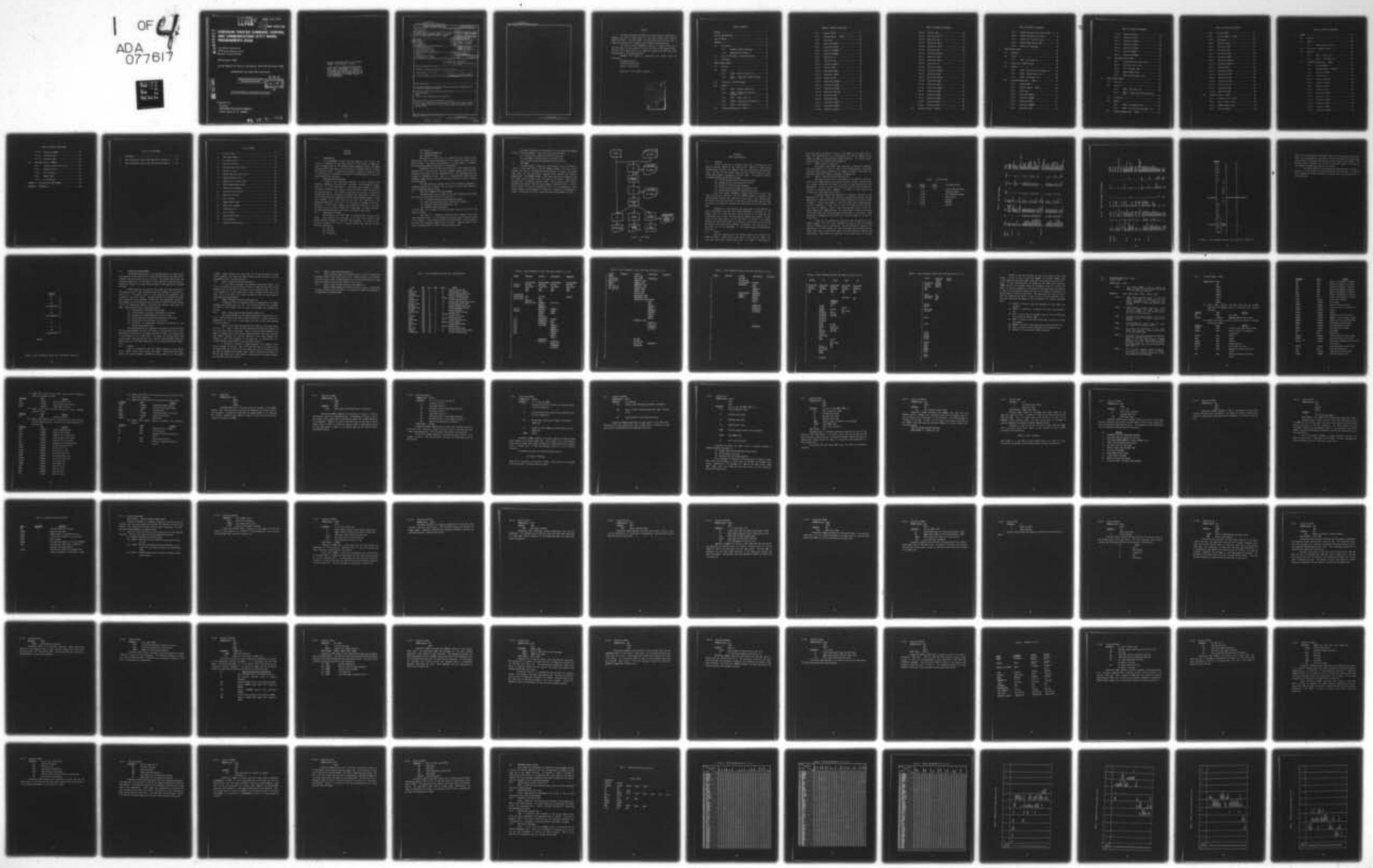
AD-A077 617 BDM CORP MCLEAN VA F/G 17/2
EUROPEAN THEATER COMMAND, CONTROL AND COMMUNICATIONS (ETC3) MOD--ETC(U)
NOV 78 P BOGUE, R LABASH, J SANDERS DNA001-78-C-0070
UNCLASSIFIED BDM/W-78-718-TR DNA-4351T-2B NL

1 OF 4
ADA
077617



DNA 4351T-2B

EUROPEAN THEATER COMMAND, CONTROL AND COMMUNICATIONS (ETC^3) MODEL PROGRAMMER'S GUIDE

The BDM Corporation
7915 Jones Branch Drive
McLean, Virginia 22102

30 November 1978

Topical Report for Period 1 December 1977—30 November 1978

CONTRACT No. DNA 001-78-C-0070

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

THIS WORK SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B363078464 O99QAXCA10903 H2590D.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, D. C. 20305

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DNA 4351T-2B	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) 6 EUROPEAN THEATER COMMAND, CONTROL AND COMMUNICATIONS (ETC^3) MODEL PROGRAMMER'S GUIDE		5. TYPE OF REPORT & PERIOD COVERED Topical Report for Period 1 Dec 77—30 Nov 78
		6. PERFORMING ORG. REPORT NUMBER 14 BDM/W-78-718-TR
7. AUTHOR(s) 10 P. Bogue R. Labash J. Sanders		8. CONTRACT OR GRANT NUMBER(s) 15 DNA 001-78-C-0070
9. PERFORMING ORGANIZATION NAME AND ADDRESS The BDM Corporation 7915 Jones Branch Drive McLean, Virginia 22102		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 Subtask 099QAXCA109-03
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D.C. 20305	11	12. REPORT DATE 30 November 1978
		13. NUMBER OF PAGES 206 17 A109
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 62704H 12 200		15. SECURITY CLASS (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

18 DNA, SBIE 19 4351T-2B, AD-E 300 625

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, ~~if different from Report~~)

18. SUPPLEMENTARY NOTES

This work sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B363078464 099QAXCA10903 H2590D.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Modeling
INCA
Nuclear Effects Communications Degradations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Draft report intended to provide the user of the European Theater Communication, Command and Control (ETC^3) computer model with computer program documentation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391 962

JOB

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

The BDM Corporation developed the European Theater Communication, Command, and Control (ETC^3) model for the INCA Program under Contract DNA001-78-C-0070. The ETC^3 model and associated data base are used to study the vulnerability of communication systems from nuclear attack.

The intent of this programmer's guide is to acquaint the user of the ETC^3 model with the program units and interfaces of the computer model. The ETC^3 user's guide serves as a companion document that describes input data preparation and program execution.

Comments and suggestions concerning this report should be directed to:

The BDM Corporation
7915 Jones Branch Drive
McLean, Virginia 22102.

Attention: Dr. William E. Sweeney, Jr.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
OVERVIEW

1-1 INTRODUCTION

This programmer's manual and the companion user's guide, reference 1 of Appendix B, have been written for the purpose of enabling a user to prepare data for the European Theater Command, Control, and Communication (ETC^3) model and to run the series of programs which will produce the required results. Information contained in these two volumes will aid in the correction of execution errors and in the modification of the software to perform new functions.

1-1.1 Document Purpose and Usage

A general level description approach is used to describe the programs, subprogram units and their interfaces. Each program or subprogram unit is briefly described by identifying the arguments or common variables, alternate entry points, and by providing a short narrative on the processing done. This enables the manual to be used as a guide to finding the detailed information available from compilation listings and not relying on detailed documentation which easily becomes outdated. Those minor changes in the code which do not alter the interfaces or the processing objectives of the unit, will not result in changes to the documentation. In some cases, however, the processing functions of the subroutines are not so easily identified or described. In these instances, the subroutine descriptions are written in considerably more detail.

1-1.2 Organization of Manual

Each program of the ETC^3 model is documented by a section in the manual. These sections are presented in an order which parallels the normal execution sequence of the ETC^3 model programs. All sections describing a program unit follow a uniform format which consists of the following subsections:

(1) Overview

(2) Input(s)

(3) Processing

(4) Output(s)

(5) Program Documentation

(6) Reference Tables

Descriptions in the first four subsections are written at the program unit level so that descriptions of the inputs and outputs are at the file level. The processing described is, in most cases, a statement of the processing objective and how it is implemented.

Subsection five provides the description of the program and subprogram units in the order of: the main program, common blocks and their variables, block data, and the subprogram units. The subprogram units are arranged in alphabetical order. Each unit is described by identifying the arguments, common blocks, alternate entry points and by providing a summary of the processing.

The last subsection presents some of the information presented in preceding subsections in tabular form for ease of use and rapid communication of subprogram relationships.

1-2 ROLE OF ETC^3 MODEL IN THE INCA PROGRAM

The BDM portion of this phase of the INCA Program had four major task areas which are identified as the following:

(1) C^3 systems assessment methodology development,

(2) Media and subsystem nuclear vulnerability analysis,

(3) European C^3 data base compilation, and

(4) Computer model development.

The ETC^3 model is the result of work performed under the last task area identified above.

The model is a computer simulation which supports the system level assessment task. It is designed to simulate large C^3 networks found in the theater. These networks consist of ground based, fixed and mobile airborne and spaceborne communication resources. Against these C^3 networks, the model plays a variety of theater nuclear events.

The model simulates the interaction of the nuclear environment with the C^3 network elements. The results are used to evaluate:

(1) the ability to complete critical C^3 functions,

(2) the impact of threat scenario variations, and

(3) C^3 network functional and physical survivability.

1-3 ETC^3 MODEL

The model proper is made up of four programs, which are Phases B, C and P of the Input Editor, and the DAMAGE program. The Input Editor programs are used in the preparation and analysis of scenario and system input data for the application programs which perform the analytic computations. The DAMAGE program is the single application program of the ETC^3 model which currently uses the Phase B and Phase C Input Editor produced Data Information Records (DIR). These program units require preprocessing support from the System Support Translator (SST). A user's manual exists for the SST program and will not be documented in this volume; however, the SST manual is included as a reference 2 of Appendix B. Figure 1 illustrates the overall processing flow at the program unit level.

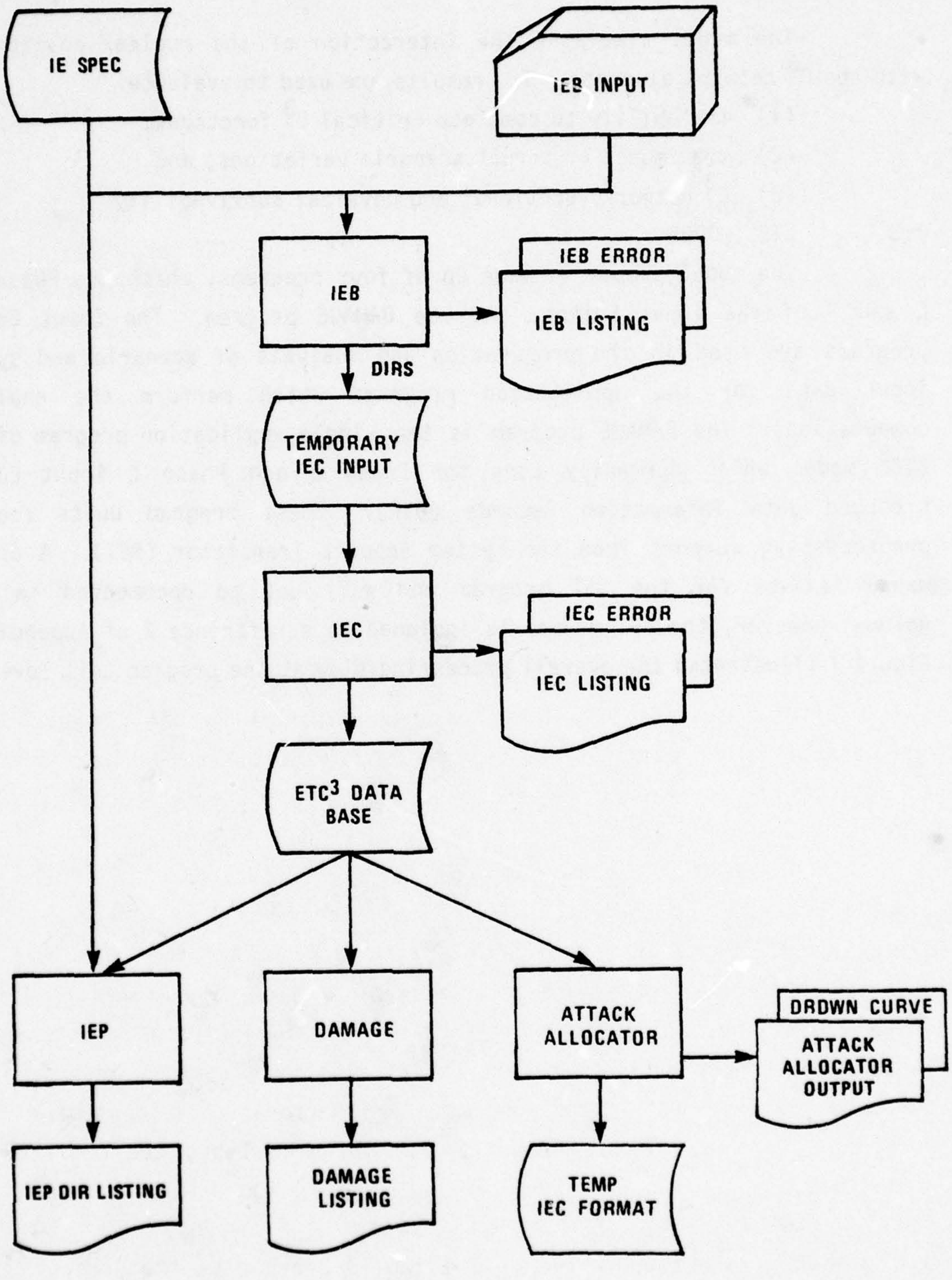


Figure 1. ETC^3 model.

SECTION 2
INPUT EDITOR PHASE B

2-1 OVERVIEW

Initial processing of scenario input data is performed by the Phase B program, IEDITB, of the Input Editor. IEDITB is a table driven program which processes the scenario data to produce Data Input Records, DIRs. The DIRs are the primary means by which scenario data are transferred from program to program in the ETC^3 model. The processing done by the IEDITB program includes the following:

(1) Initialize data entries to specified values,

(2) Assemble blocks of information for a DIR,

(3) Search for associated data elements,

(4) Initiate subroutine processing functions, and

(5) Assemble DIRs and add DIR identification data.

Since the program implements a dynamic table driven process, the Table entries are read each time the IEDITB program is executed. The IEDITB program requires two input data sources; one is the scenario data, the other is the specifications for the table. Specification data are used to control the processing sequence which transforms the scenario data into DIRs.

Preparation of the scenario and system data for the IEDITB program is described in the ETC^3 User's Manual Reference 1 of Appendix B. The syntax rules for coding the data and structuring of the data deck are described as well as the interrelationships between data elements. Preparation of the specification file for the table is included in this manual because the specification data are essential to the control of processing performed by the Input Editor programs. Furthermore, this input data is of much more use to the programmer in the program debug and modification activities.

2-2 INPUTS

Data are transferred to the IEDITB program by the logical files TAPE5 and TAPE11. TAPE5 contains the scenario input data and TAPE11 contains the Input Editor specification data. The former is maintained in

card form, while the latter is kept in card image form on either disk or tape. The specification file is maintained by using the UPDATE utility system as a means to maintain model program integrity. This permits specification files to be traceable and reproducible.

2-2.1 TAPE5 - Scenario Input File

The scenario and system input data deck is copied to a temporary disk file before it is used as input to the IEDITB program. The syntax rules and card coding conventions for preparing scenario input data are described in the user's manual, reference 1 of Appendix B.

2-2.2 TAPE11 - Input Editor Specification File

The specification file is the product of an UPDATE run and consists of card image records which specify the table values. The specification records, identified as Z-cards, are used for data description or DIR description. The way in which the Z-cards are used for data and DIR description is outlined below. Appendix A provides the card formats and explanations of the specification cards.

The table contains two categories of information necessary to the generation of DIRs. These are identified respectively as data and DIR descriptions. Data descriptions are prepared according to the card format of Table 1. The ZD card format is the only format used to code data description information. A ZEND card is used to indicate that data description input is complete. Table 2 shows the data description values currently used by the IEDITB program. The values shown in Table 2 are unique to the ETC^3 model and are not changed from execution to execution. These values may be changed by altering the set of ZD card images in the specification file.

There are two methods for specifying DIR descriptions to the IEDITB program. The first form, indicated in Figure 2, uses a ZF format card to introduce the DIR description. The DIR description consists of a combination of Z-card formats for specifying data forms and data blocks. Either data forms or data blocks may be used individually to specify DIR descriptions. Data blocks are specified in the same manner as the data

Table 1. ZD card format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZD	Z-Card Type Designator
2	5-10		Reference Number
3	11-20		Units Reference Number
4	21-30		Scale Factor
5	31-40		Minimum
6	41-50		Maximum
7	51-60		Quantum

Table 2. Input data summary.

INPUT SECTION	DATA NAME (DN)	DN Ab.	DATA TYPE	DEFAULT	MINIMUM	MAXIMUM	QUANTUM	PHYSICAL UNITS
CONTROL	TITLE		CHAR. STRING	BLANK	0	80	1	CHARACTER
	CLASS		CHAR. STRING	BLANK	0	28	1	CHARACTER
	TERRAIN		SPECIFIC NAME	BLANK	0	16	1	CHARACTER
THREATCHAR	IDENT	ID	USER-NAME	USER INPUT	0	16	1	CHARACTER
	TYPE		SPECIFIC NAME	BM	-	-	-	BM, ICBM, SLBM
	YIELD		NUMERIC	1.0	0	26843.5455	.0001	KILOTON
	CEP		NUMERIC	0	0	16383	1	METER
	PROBVEHICLE	PV	NUMERIC	1.0	0	1	.01	UNITLESS
SITECHAR	IDENT	ID	USER-NAME	USER INPUT	0	16	1	CHARACTER
	TYPE		SPECIFIC NAME	COM	0	16	1	COM, RELAY, C2, C3, AIRBASE, TERM
	OPKILL	OPK	NUMERIC	6553.5	0.1	6553.4	0.1	POUNDS/IN^2
	DPKILL	DPK	NUMERIC	6553.5	0.1	6553.4	0.1	POUNDS/IN^2
	THERMALKILL	THK	NUMERIC	255.0	1.0	254.0	1.0	CALORIES/CM^2
	GAMMAKILL	GMK	NUMERIC	65535	1.0	65534.	1.0	RADS
	GAMMADOTKILL	GMDK	NUMERIC	0	10^5	10^{14}	-	RADS/SEC
	NEUTRONKILL	NK	NUMERIC	0	1	1×10^{35}	-	NEUTRONS/CM^2
	HAEMPKILL	HEK	NUMERIC	102.3	.1	102.2	0.1	KILOVOLTS/METER
	LAEMPKILL	LEK	NUMERIC	102.3	.1	102.2	0.1	KILOVOLTS/METER
	XRAYKILL	XRK	NUMERIC	10.23	10^{-6}	10.22	10^{-6}	CAL/CM^2
	RADIATIONKILL	RDK	NUMERIC	5110.	10.	5100.	10.	REMS
	OPIMPULSEKILL	OPIK	NUMERIC	6553.5	0.1	6553.4	0.1	POUNDS-SECS/IN^2
	DPIMPULSEKILL	DPIK	NUMERIC	6553.5	0.1	6553.4	0.1	POUNDS-SECS/IN^2
	VULNUMBERKILL	VNK		99P0 99Q0	1x0	57x9		UNITLESS
	PROBDAMAGEVNK	PDVNK	NUMERIC	.90	0.01	1.00	0.01	UNITLESS
	ACOUSTICKILL	AK	NUMERIC	1023.0	1.0	1022.	1.0	DECIBELS/μPASCAL (1YD)
	IDMESSAGECHAR	IDMC	USER-NAME					
	IDPROCEDURE	IDP	USER-NAME	MP9999				
COMSYSTEMCHAR	IDENT	ID	USER-NAME	USER INPUT	0	16	1	CHARACTER
	TYPE		SPECIFIC NAME	CABLE	0	16	1	CABLE, MICROWAVE, ELF, VLF, LF, HF, VHF, UHF, SAT
	OPKILL	OPK	NUMERIC	6553.5	0.1	6553.4	0.1	POUNDS/IN^2
	DPKILL	DPK	NUMERIC	6553.5	0.1	6553.4	0.1	POUNDS/IN^2
	THERMALKILL	THK	NUMERIC	255.0	1.0	254.0	1.0	CALORIES/CM^2
	GAMMAKILL	GMK	NUMERIC	65535.	1.0	65534.	1.0	RADS
	GAMMADOTKILL	GMDK	NUMERIC	0	10^5	10^{14}	-	RADS/SEC
	NEUTRONKILL	NK	NUMERIC	0	1	10^{35}	-	NEUTRONS/CM^2
	HAEMPKILL	HEK	NUMERIC	102.3	.1	102.2	0.1	KILOVOLTS/METER
	LAEMPKILL	LEK	NUMERIC	102.3	.1	102.2	0.1	KILOVOLTS/METER
	XRAYKILL	XRK	NUMERIC	10.23	10^{-6}	10.22	10^{-6}	CAL/CM^2
	OPIMPULSEKILL	OKIK	NUMERIC	6553.5	0.1	6553.4	0.1	POUNDS-SEC/IN^2
	DPIMPULSEKILL	DPIK	NUMERIC	6553.5	0.1	6553.4	0.1	POUNDS-SEC/IN^2
	VULNUMBERKILL	VNK		99P0.99Q0	1x0	57x9		UNITLESS
	PROBDAMAGEVNK	PDVNK	NUMERIC	.90	0.01	1.00	0.01	UNITLESS
	FREQUENCY	FREQ	NUMERIC	0	0	1677.215	0.001	MEGAHERTZ
	CHARRATE	CR	NUMERIC	1677.7215	0	1677.7215	.0001	CHARACTERS/SEC.
	EQUIPMENT		SPECIFIC-NAME					LINE, NODE 1, NODE 2

Table 2. Input data summary (continued).

MESSAGECHAR	IDENT	ID	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	CHARACTER	CHAR	NUMERIC	100	0	16777215	1	UNITLESS
PROCEDURE	IDENT	ID	USER-NAME	USER INPUT	0	16	1	CHARACTERS
SITE	IDENT	ID	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	LOCATION	LOC		USER INPUT				
	LEVEL		NUMERIC	1	0	255	1	UNITLESS
	TERMINAL CODE	TMC	SPECIFIC-NAME	BLANK	0	16	1	A,D,S,E,R,B,C,T,L,M,V,P,L.
	IDSITECHAR	IDC	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	IDWEAPON	IDW	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	HEIGHT OF BURST	HOB	NUMERIC	-2×10^7	0	16777215	1	METERS
COMLINE	IDENT	ID	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	TYPE		SPECIFIC-NAME	FULL				ONEWAY, HALF, FULL
	NORMOPSNRATIO	SNR	NUMERIC	20	0	31	1	DECIBELS
	PROBOPBENIGN	POB	NUMERIC	1.0	0.0	1.0	.001	UNITLESS
	CAPACITY	CAP	NUMERIC	100	0	10^9	1	ASSIGNED BY USER
	CONNECT		USER-NAME	USER INPUT	0	16	1	CHARACTERS
	IDCOMSYSCHAR	IDCSC	USER-NAME	BLANKS	0	16	1	CHARACTERS
COMNET	IDENT	ID	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	COMLINE	LINE	USER-NAME	USER INPUT	0	16	1	CHARACTERS
EVENT	IDENT	ID	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	TYPE		SPECIFIC-NAME	USER INPUT	0	16	1	DEBUG, NODEBUG, ENDGAME, ABORT, MSG. MESSAGE
	TIME		NUMERIC	USER INPUT	0	*MAXTIME	*ONTIME	SECONDS
	IDSITExxx	IDSxxx	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	IDCOMLINE	IDCL	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	IDCOMNET	IDCN	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	IDMESSAGECHAR	IDMC	USER-NAME	USER INPUT	0	16	1	CHARACTERS
THREATBM	IDENT	ID	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	VELOCITY	V	NUMERIC	1000	.1	6553.5	0.1	METERS/SEC
	ANGLELAUNCH	AL	NUMERIC	90	0	89.9	0.0057	DEGREES
	NUMOBJECTS	NOC	NUMERIC	1	0	7	1	UNITLESS
	IDTHREATCHAR	IDC	USER-NAME	USER INPUT	0	16	1	CHARACTERS
	IDLAUNCH	IDLNCH	USER-NAME	BLANKS	0	16	1	CHARACTERS
	IDTARGET	IDTGT	USER-NAME	BLANKS	0	16	1	CHARACTERS
	TIMELAUNCH	T1	NUMERIC	0	0	268435.46	0.01	SECONDS
	TIMEBURST	T2	NUMERIC		70	268435.46	0.01	SECONDS
	LATBURST	F2	NUMERIC	41.5	-89	89	0.00057	DEGREES
	LONGBURST	G2	NUMERIC	87.5	-180	180	0.00057	DEGREES
	ALBURST	H2	NUMERIC	0	0	100000	1	METERS
	LATLAUNCH	F1	NUMERIC	60	-89	89	.00057	DEGREES
	LONGLAUNCH	G1	NUMERIC	-80	-180	180	.00057	DEGREES
	ALTLAUNCH	H1	NUMERIC	0	0	32767	1	METERS

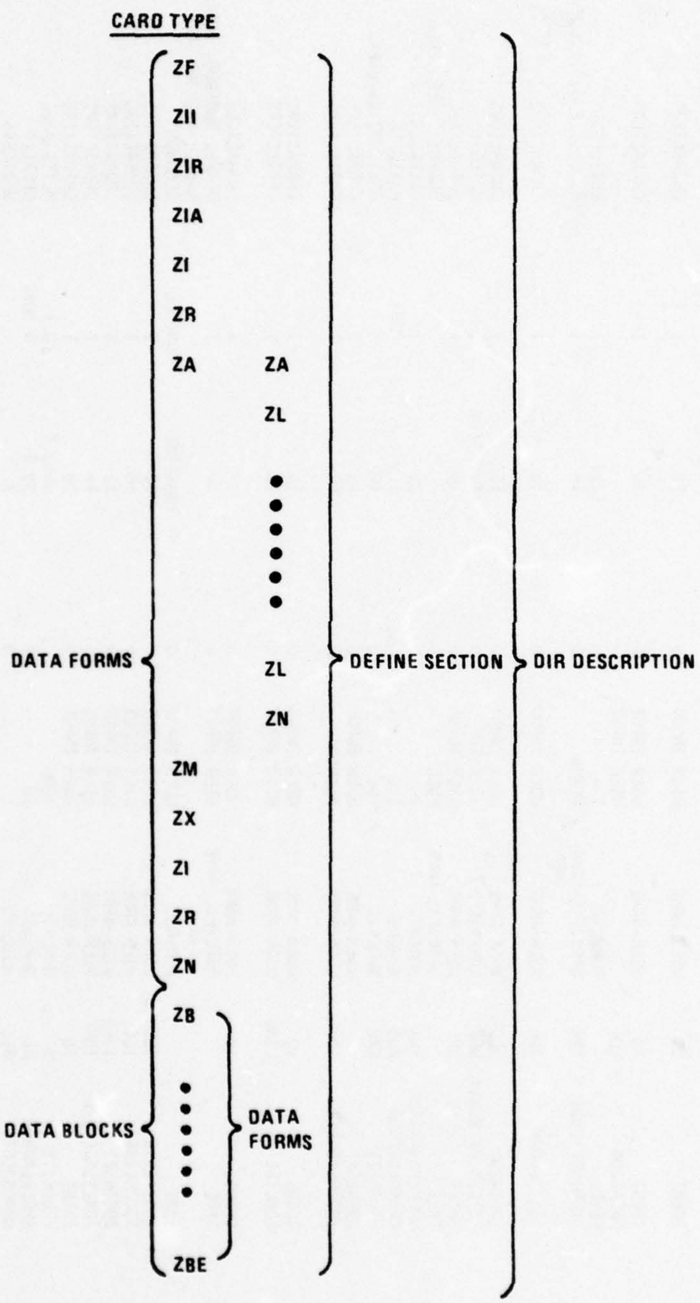


Figure 2. Data information record, DIR, description (method 1).

forms, but exist between ZB and ZBE format cards to indicate that the data form is to be treated as a data block. The ZM card format and an alternate form of the ZA card format introduce one or more other Z-card formats. The ZM card format introduces the ZX, ZI, and ZR card formats, while the ZA card format introduces the single ZL card format.

The second method of introducing DIR specifications is shown in Figure 3. In this method, the define sections which are shown in Figure 2 are introduced by ZT and ZFD cards to indicate that a DIR specification is to be processed.

CARD TYPE

ZF

ZT

ZFD

[DEFINE SECTION]

ZT

ZFD

[DEFINE SECTION]

•
•
•
•

ZT

ZFD

[DEFINE SECTION]

5610/78W

Figure 3. Data information record, DIR, description (method 2).

2-3 PROCESSING-IEDITB PROGRAM

The processing objective of the IEDITB program is to organize the user prepared scenario data into a structured data base so that application programs requiring access may do so easily. The IEDITB program does the necessary conversions, initializations, and validations of the variables to enable the transfer of correct and valid data to the application programs via DIRs.

IEDITB processes the scenario and system input data according to the syntax rules for input preparation and the directives of the specification file. The syntax rules for the scenario data preparation are contained in the user's manual. The specification file indicates how input data values are to be converted, initialized, and represented internally. The program uses the processing codes and pointers within the specification file to do operations needed to transform the scenario input data into DIRs. Some functions done by the program include:

(1) Construction of DIR identification data,

(2) Initialization, conversion, and typing of variables,

(3) Searching for associated data for a given input

(4) Storing such data into specified locations of the DIR,

(5) Concatenation of data blocks to a DIR, and

(6) Initiation of specialized subroutine processing on the scenario and system input.

The constructed DIRs will represent the scenario and system input data by being expanded in such a way that all required data for an event described by a DIR will be included specifically either as an entry value or be pointed to by an index value. The syntax for coding scenario and system inputs, the specification file, and the IEDITB program enable the description of an ETC^3 model data base in a user comprehensible and compact form.

2-4 OUTPUTS

Data is transferred from the IEDITB program via the logical files: TAPE2, TAPE6, TAPE7, TAPE12, and TAPE13. TAPE6 prints the informative and error messages accompanying each execution of the IEDITB

program. TAPE7 produces the binary DIRs for use by the other two Input Editor programs, and TAPE2 and TAPE13 produce printed results to aid the programmer or user in correcting errors.

2-4.1 TAPE2 - Scenario and System Input File

All the data card images from TAPE5 are copied onto TAPE2. This file is printed by using job control cards to reposition the file to the beginning and to copy the file contents to the system's OUTPUT file. The file contents are in card image form so that a system utility copy can be used. The file contents are normally printed to show the exact scenario input data and input sequence.

2-4.2 TAPE6 - Informative and Error Messages File

Informative and error messages, which are generated by the IEDITB program, are printed from this file. The system's OUTPUT file is equated to TAPE6 on the program card so that all messages are automatically printed.

2-4.3 TAPE7 - Binary Data Information Record (DIR) File

The DIRs generated by the IEDITB program are transferred by this file. TAPE7 consists of the DIRs in binary form and is normally saved as a permanent file. This file is the primary DIR input to the other programs of the Input Editor.

Tables 3 and 4 show the structure and contents of the DIRs making up this file. Table 3 shows the DIR identification data which appear as the first four words of each DIR. Table 4 shows the data elements which make up a DIR for each of the section names described in the user's manual. Both tables identify the relative location of the data elements with respect to other data elements in the same DIR. A complete DIR will include the four words of Table 3 and the words of Table 4.

A DIR may be variable in length because of the number of data blocks included in the DIR. Each block type which may be included will be of fixed length and structure but the number and types of blocks included per DIR will depend on the scenario data. Table 4 shows how the words of a DIR are associated with a data element. Explanations of the data elements are included in the user's manual.

2-4.4 TAPE12 - Specifications Debug File

TAPE12 produces debug print material which is used in identifying problems related to the specification file. TAPE12 contents are printed by using job control cards to reposition the file to the beginning and to copy the file contents to the system's OUTPUT file for print.

2-4.5 TAPE13 - Data Information Record, DIR, Debug File

TAPE13 produces debug print material which is used in identifying problems related to the DIR output. The contents of TAPE13 are produced in the same way as for TAPE12 above.

Table 3. Data information record, DIR, identification.

DIR	MFMT	MTYP	MCONT	MLEN(?)	PURPOSE
Control	1024	0	0	47	Identifies the data base
Control Count	1024	10	0	14	Summarize number of records
Threatchar	1040	11	0	13	Describes threat characteristics
Threatchar Count	1040	10	0	6	Provides Threatchar DIR count
Sitechar	1040	21	0	Variable	Describes site charcteristsics
Sitechar Count	1040	20	0	6	Provides sitechar DIR count
Comsyschar	1040	31	0	113	Describes Comm. systems characteristics
Comsyschar Count	1040	30	0	6	Provides Comsyschar DIR count
Messagechar	1040	41	0	10	Describes Message characteristics
Messagechar Count	1040	41	0	6	Provides messagechar DIR count
Procedure	1040	51	0	9	Decribes procedures
Procedure Count	1040	50	0	6	Provides Procedure DIR count
Site	1040	101	0	Variable	Describes Sites
Site Count	1040	102	0	6	Provides Site DIR count
Comline	1040	111	0	Variable	Describes Comlines
Combine Count	1040	110	0	6	Provides Comline DIR count
Comnet	1040	121	0	Variable	Describes Comnets
Comnet Count	1040	120	0	6	Provides Comnet DIR count
Event	1040	131	0	Variable	Describes Events
Event Count	1040	130	0	6	Provides Event DIR Count
Threatbm	1100	222	0	48	Describes ballistic missile threats
Threatbm Count	1100	220	0	6	Provides threatbm DIR Count.

Table 4. Data information record, DIR, word structure (1 of 5).

	CONTROL	THREATCHAR	SITECHAR	COMSYSTEMCHAR	MESSAGECHAR
1					
2					
3		RELATIVE NUMBER	RELATIVE NUMBER	RELATIVE NUMBER	RELATIVE NUMBER
4	DATE(IEB)	NEXT NAME	NEXT NAME	NEXT NAME	NEXT NAME
5	TIME(IEB)	FIXED LENGTH	FIXED LENGTH	FIXED LENGTH	FIXED LENGTH
6		IDENT	IDENT	IDENT	IDENT
7		IDENT	IDENT	IDENT	IDENT
8					
9	CLASSIFICATION				
10	CLASSIFICATION	YIELD	SITE TYPE		CHARACTER
1	CLASSIFICATION	CEP	OPKILL		
2		PROGVEHICLE	OPKILL		
3		THREAT VEH. TYPE	THERMALKILL	CIRCUIT TYPE	
4			GAMMAKILL		
5			GAMMADOTKILL		
6	DATE(IEG)		NEUTRONKILL	FREQUENCY	
7	TIME(SEC)		HAEMPKILL	CHARRATE	
8			LAEMPKILL	CHANNEL	
9			XRAYKILL		
20			RADIATIONKILL		
1	TITLE		OPIMPULSEKILL	EQUIPMENT	
2	TITLE		DPIMPULSEKILL	OPKILL	
3	TITLE		VULNUMBERKILL	OPKILL	
4	TITLE		VN K-FACTOR	THERMALKILL	
5	TITLE			GAMMAKILL	
6	TITLE			GAMMADOTKILL	
7	TITLE			NEUTRONKILL	
8	TITLE			HAEMPKILL	
9				LAEMPKILL	
30				XRAYKILL	
1			PROGDAMAGEVNK		
2			ACOUSTICKILL	OPIMPULSEKILL	
3				DPIMPULSEKILL	
4				VULNUMBERKILL	
5				VN K-FACTOR	
6					
7					
8					
9					
40					

Table 4. Data information record, DIR, word structure (2 of 5).

	CONTROL	THREATCHAR	SITECHAR	COMSYSTEMCHAR	MESSAGECHAR
1	TERRAIN		OPKILL SIGMA		
2	UTM GRID ZONE		OPKILL SIGMA	PROBDAMAGEVNK	
3	ORIGIN LETTERS		THERMALKILL SIGMA		
4	ORIGIN X		GAMMAKILL SIGMA		
5	ORIGIN Y		GAMMADOTKILL SIGMA		
6	VERT. DATA RATE		NEUTRONKILL SIGMA		
7	LAT. DATA RATE		HAEMPKILL SIGMA		
8			LAEMPKILL SIGMA		
9			XRAYKILL SIGMA		
50			RADIATIONKILL SIGMA		
1			OPIMPULSEKILL SIGMA		
2			OPIMPULSEKILL SIGMA	EQUIPMENT	
3				OPKILL	
4				DPKILL	
5				THERMALKILL	
6				GAMMAKILL	
7				GAMMADOTKILL	
8				NEUTRONKILL	
9				HAEMPKILL	
60				LAEMPKILL	
1				XRAYKILL	
2			ACOUSTICKILL SIGMA		
3				OPIMPULSEKILL	
4				OPIMPULSEKILL	
5				VULNUMBERKILL	
6				VN K-FACTOR	
7					
8					
9					
70					
1			NEXT NAME		
2			TYPE MSG CHAR		
3			RELATIVE NO.	PROBDAMAGEVNK	
4			IDMESSAGECHAR		
5					
6					
7					
8					
9					
80					

Table 4. Data information record, DIR, word structure (3 of 5).

	CONTROL	THREATCHAR	SITECHAR	COMSYSTEMCHAR	MESSAGECHAR
1			NEXT NAME		
2			TYPE MSG PROC		
3			RELATIVE NUMBER	EQUIPMENT	
4			IDPROCEDURE	OPKILL	
5				DPKILL	
6				THERMALKILL	
7				GAMMAKILL	
8			NUMBER ARGUMENTS	GAMMADOTKILL	
9			ARGUMENT 1	NEUTRONKILL	
90			ARGUMENT 2	HAEMPKILL	
1				LAEMPKILL	
2				XRAYKILL	
3					
4				OPIMPULSEKILL	
5				DPIMPULSEKILL	
6				VULNUMBERKILL	
7				VN K-FACTOR	
8					
9					
100					
1					
2					
3					
4				PROBDAMAGEVNK	
5					
6					
7					
8					
9					
110					
1					
2					
3					
4					
5					
6					
7					
8					
9					
120					

Table 4. Data information record, DIR, word structure (4 of 5).

	PROCEDURE	SITE	COMLINE	COMNET	EVENT
1					
2					
3	RELATIVE NUMBER	RELATIVE NUMBER	RELATIVE NUMBER	RELATIVE NUMBER	RELATIVE NUMBER
4	NEXT NAME	NEXT NAME	NEXT NAME	NEXT NAME	NEXT NAME
5	FIXED LENGTH	FIXED LENGTH	FIXED LENGTH	FIXED LENGTH	FIXED LENGTH
6	IDENT	IDENT	IDENT	IDENT	IDENT
7	IDENT	IDENT	IDENT	IDENT	IDENT
8					
9					
10				COUNT COMLINES	TYPE
1					TIME
2			NORMOPSNRATIO		
3			PROBOPBENIGN		
4			CAPACITY		
5		LATITUDE	TYPE		
6		LATITUDE		NEXT NAME	
7		LONGITUDE	NEXT NAME	TYPE	
8		LONGITUDE	TYPE	RELATIVE NUMBER	
9		ALTITUDE(MSL)	RELATIVE	LINE	
20		ALTITUDE(HAT)	IDENT NODEI		
1		UTM LOCATION	IDENT NODEI		
2		UTM LOCATION			
3		UTM LOCATION			
4		UTM LOCATION			
5		EASTING			
6		EASTING	NEXT NAME		
7		NORTHING	TYPE CODE		
8		NORTHING	RELATIVE NUMBER		
9		LEVEL	IDENT NODE 2		
30		TERMINALCODE	IDENT NODE 2		
1		NEXT NAME			
2		C3CHAR TYPE			
3		RELATIVE NUMBER			
4		IDCHAR			
5		IDCHAR	NEXT NAME		
6			TYPE		
7			RELATIVE NUMBER		
8			IDCSC		
9		NEXT NAME	IDCSC		
40		THREATCHAR TYPE			
1		RELATIVE NUMBER			
2		IDWEAPON			
3		IDWEAPON			
4					
5					
6		HEIGHTOFBURST			
7					
8					
9					

Table 4. Data information record, DIR, word structure (5 of 5).

	THREATBM	CONTROL COUNT	COUNT(AU)
1		THREATCHAR	(SECTION)
2		SITECHAR	
3	RELATIVE NUMBER	COMSYSCHAR	
4	NEXT NAME	MESSAGECHAR	
5	FIXED LENGTH	PROCEDURE	
6	IDENT		
7	IDENT		
8			
9			
10	VELOCITY	SITE	
1	ANGLELAUNCH(INA)	COMLINE	
2	ANGLELAUNCH(A)	COMNET	
3		EVENT	
4			
5			
6	NUMOBJECTS		
7	NEXT NAME		
8	TYPE		
9	RELATIVE NUMBER		
20	IDTHREATCHAR		
1			
2			
3			
4	IDLAUNCH		
5			
6			
7			
8	IDTARGET		
9			
30			
1			
2			
3	LATLAUNCH		
4	LATLAUNCH		
5	LONGLAUNCH		
6	LONGLAUNCH		
7	ALTLAUNCH		
8			
9	TIMELAUNCH		
40	TIMELAUNCH		
1	LATBURST		
2	LATBURST		
3	LONGBURST		
4	LONGBURST		
5	ALTBURST		
6			
7	TIMEBURST		
8	TIMEBURST		
9			
50			

IEDITB is the main program for the first phase of the Input Editor. It sets the size of the blank COMMON. The variable IADMAX is set to the length of the blank common array IADS to be used less one entry of 7 words. If more specifications are added and the IADS table overflows, IADMAX and the field length should be increased in order to run the program. As the main program for this phase of the Input Editor processing, it acts as the executive routine and calls on a number of subroutines to perform special processing functions. The subroutine calls and call structure are summarized by Tables 12 and 13 of 2-6, but the subroutine calls and the processing performed by the subroutines are briefly described below.

(1) ERRSET is called to make the occurance of error number 150 nonfatal.

(2) UNITST is called to initialize file units and pagination data.

(3) GET is called with an argument value of zero to cause the first record on TAPE5 to read.

(4) PAGE is called at the entry point PAGEC to print the input card images.

(5) READDD is called to read and process the specification file.

(6) PAGE is called to control pagination on the output files.

(7) AUTOPB is called to process TAPE5 input data.

2-5 PROGRAM DOCUMENTATION - IEDITB

2-5.1 Program IEDITB

COMMON BLOCK: ZZDST

ZZDST - The variable IADMAX is used to indicate the number of words in blank common (ADS array) It is set within this program.

ARGUMENTS: TAPE2, TAPE5, TAPE7, TAPE11, TAPE12, TAPE13

TAPE2 - TAPE2 contains printed output. All the card images from TAPE5 are copied to this file, and is processed with a REWIND and COPY (TAPE2, OUTPUT) to cause the contents to be printed.

TAPE5 - TAPE5 contains scenario input data. These are card images, with data contained in columns 1 through 72. The user must insure this file is set up correctly.

TAPE7 - An output file containing DIRs. This file is normally passed to the Phase C Input Editor program.

TAPE6 - Error messages are output to this file, which is currently equivalenced to OUTPUT.

TAPE11 - The input file containing all the Input Editor specifications. These specifications are in card image format.

TAPE12 - An output file containing debug information concerning program operation. If the contents of this file are needed, a REWIND (TAPE12) and a COPY (TAPE12, OUTPUT) should be placed in the JCL. This file should not be printed unless there are problems with the specifications.

TAPE13 - This file also contains debug information. It is used in a manner similar to TAPE12. This file should not be printed unless there are problems with the DIR output.

2-5.2 Program Commons - IEDITB

COMMON BLOCKS: BLANK
ZZB
ZZBLK
ZZCVX
ZZDST
ZZEDCT
ZZFILA
ZZPS
ZZUNIT

(1) Blank common contains the IADS table and the variable specifying the maximum length of the array which contains the table.

VARIABLE	TYPE	CONTENT
ADS	REAL	IADS TABLE
IAD MAX	INTEGER	MAXIMUM LENGTH OF IADS TABLE

(2) Common ZZB contains various conversion factors, constants and system counters.

VARIABLE	TYPE	CONTENT
SECNAM	Real	Scenario input data section name
PACKRD	Real	Round off level of geographic coordinates
PACKR	Real	Unused
TMAX	Real	Unused
TQUANT	Real	Unused
DBLANK	Real	Blank Characters
DZERO	Real	Double precision zero
D2R	Real	Degrees to radians conversion factor
R2D	Real	Radians to degrees converison factor

VARIABLE	TYPE	CONTENT
R15	Real	Angle of 15 degrees in radians
R30	Real	Angle of 30 degrees in radians
R45	Real	Angle of 45 degrees in radians
R60	Real	Angle of 60 degrees in radians
R90	Real	Angle of 90 degrees in radians
R180	Real	Angle of 180 degrees in radians
R360	Real	Angle of 360 degrees in radians
LERR	Integer	Error level indicator
NERR	Integer	Error Count
LERRP	Integer	Level of errors to print
LERRS	Integer	Level of error to terminate run
MSCNT	Integer	Unused
NVEHA	Integer	Unused
NVEHB	Integer	Number of ballistic missiles
NERRA	Integer	Number of errors found by PTHVA
NERRB	Integer	Number of errors found by PTHVB
MDMFT	Integer	DIR format being processed
MDTYP	Integer	DIR type being processed
MDCNT	Integer	DIR continuation number
MDLEN	Integer	DIR length
CD	Real	DIR buffer
MD2FMT	Integer	DIR continuation record format
MD2TYP	Integer	DIR continuation record type
MD2CNT	Integer	Continuation number of DIR continuation record
MD2LEN	Integer	DIR continuation record length
CD2	Real	Alternate DIR buffer
ND12	Integer	Continuation record offset
LZONE	Integer	UTM zone letter in DIR

(3) Common ZZBLK contains data which describe the information data blocks added to the DIR.

VARIABLE	TYPE	CONTENT
DIR	Logical	Force ner DIR if true
BLKPT	Integer	Data block location pointer
KODE	Integer	Code number of block

(4) Common block ZZCVX contains conversion factors between systems of units.

VARIABLE	TYPE	CONTENT
CVX	Real	Conversion factors

(5) Common block ZZDST contains information about the format of the DIR specifications, and the array containing the DIR specification.

VARIABLE	TYPE	CONTENT
IBDD	Integer	Index to begin ZD data
ILDD	Integer	Index to last ZD data
IQDD	Integer	Quantum for a ZD data block
IBSS	Integer	Index to begin Z card data
ILSS	Integer	Index to last Z card data
IQSS	Integer	Quantum for Z card data block
IXREF	Integer	ZD card reference field
IXUNIT	Integer	ZD card unit field
IXSCAL	Integer	ZD card scale field
IXMIN	Integer	ZD card minimum field
IXMAX	Integer	ZD card maximum field
IXQNTM	Integer	ZD card quantum field
IXTYPE	Integer	Card type field
IXNAME	Integer	Card name field
IXDA	Integer	Card "A" field
IXDB	Integer	Card "B" field
IXDC	Integer	Card "C" field
IXDD	Integer	Card "D" field

(6) Common block ZZPS is not being used by this version of the Input Editor program.

(7) Common block ZZPS contains input stream descriptive data.

VARIABLE	TYPE	CONTENT
LEVEL	Integer	Parentheses depth counter
KIND	Integer	Identifier of current token
CBUF	Real	Warrant token
KIND9	Integer	Parantheses depth, next token
KIND2	Integer	Identify next token
CBUF2Q	Real	Next token

(8) Common block ZZUNIT contains constants and conversion factors.

VARIABLE	TYPE	CONTENT
U1	Real	Radius of earth in meters
U2	Real	Pi divided by 2
U3	Real	Degrees to radians conversion factor
U4	Real	Double precision zero
U5	Real	Radians to degrees conversion factor

2-5.3 Block Data

COMMON BLOCKS: ZZB

ZZFILA

ZZDST

ZZUNIT

The BLOCK DATA unit initializes values of variables in the common blocks. The listing should be consulted to determine the values used for the variables. It should be noted that the common ZZUNIT is not from the SST library, therefore care should be taken in modifying initialization values for variables in ZZUNIT.

2-5.4 Subroutine ALOBLK

COMMON BLOCKS: ZZB

ZZBLK

ZZDST

ARGUMENT: BIAS

BIAS	Array index for DIR data block's first word

Subroutine ALOBLK manages the allocation of space in a DIR or continuation record for a DIR data block. It determines if there is available space in the current DIR or continuation record for the DIR data block to be appended. ALOBLK will place a DIR data block on the DIR or continuation record if there is sufficient space or start a continuation record for the data block if there is not enough space on the current DIR or continuation record.

2-5.5 Subroutine ARHLOC

COMMON BLOCK: ZZUNIT

ARGUMENTS: F1, G1, H1, F2, G2, H2, AZ, RG

F1	Latitude of point 1
G1	Longitude of point 1
H1	Altitude of point 1 above mean sea level
F2	Latitude of point 2
G2	Longitude of point 2
H2	Altitude of point 2 above mean sea level
AZ	Azimuth of point 1 with respect to point 2
RG	Slant range from point 1 to point 2

ENTRY POINT: LOCARH

Subroutine ARHLOC computes the latitude and longitude of point 1 given the slant range and azimuth from point; and the latitude and longitude of the second point. The altitudes of both points are also given.

Entry point LOCARH computes the azimuth and range of point 2 with respect to point 1 from the altitude, latitude, and longitude values of both points.

2-5.6 Subroutine AZRNGE

COMMON BLOCK: ZZUNIT

ARGUMENTS: F1, F2, RG, AZ, DUMMY

F1	Array containing the latitude and longitude values for point number one
F2	Array containing the latitude and longitude values for point number two
RG	Ground range between point number one and point number two
AZ	Azimuth of point number two with respect to point number one
DUMMY	Unused

Subroutine AZRNGE computes the ground range and azimuth between point number one and point number two. Point number one is the reference point and the range and azimuth are measured in meters and radians respectively. Azimuth zero is north and measured positive in the clockwise direction.

The expression used to calculate the ground range is:

$$RG = 2R_e \sin^{-1}(RG/2R_e)$$

where Re is the radius of the earth in meters. Both points are considered to be on the earth's surface at mean sea level.

2-5.7 Subroutine BOUNDS

COMMON BLOCK: ZZDST

ARGUMENTS: ILB, IUB, KW

ILB Index of data item needing boundary information

IUB Index of data item preceding data item indicated by KW

KW Type indicator of the terminating entry

Subroutine BOUNDS establishes an upper bound in the IADS array. Starting at the index specified by ILB, the IADS table is checked for a DIR specification corresponding to the type specified by KW.

2-5.8 Subroutine CK

COMMON BLOCKS: ZZCVX
ZZB
ZZPS

ARGUMENTS: ICV, CL, CU, CQ, NAME, INDEX, LE

ICV	Units conversion index
CL	Minimum value index
CU	Maximum value index
CQ	Quantum value index
NAME	Array of names printed in error messages
INDEX	Array NAME index
LE	Error level indicator

Subroutine CK checks the input value of a numeric variable to determine whether the number is:

(1) between specified maximum and minimum values,

(2) of the proper units, and

(3) expressed with the proper quantum.

Error messages are produced by the subroutine to identify inputs which have values greater or lesser than the maximum and minimum limits values, respectively. The message will identify the data section, data name, input value, the limits and the error level of the incorrectly specified variable value.

2-5.9 Subroutine CKD

COMMON BLOCKS: ZZB

ZZCVX

ZZPS

ARGUMENTS: ICV, CL, CU, CQ, NAME, INDEX, LE

ICV	Units conversion index
CL	Minimum value index
CU	Maximum value index
CQ	Quantum value index
NAME	Array of names printed in error messages
INDEX	Array NAME index
LE	Error level indicator

ENTRY POINT CKP

Subroutine CKD is included to maintain compatibility with the IBM version of the Input Editor. In the CDC version, the routine performs the same functions as subroutine CK. Subroutine CK checks numeric data against maximum and minimum values and does unit conversions and quantization of the variable value.

Entry point CKP uses hard coded values for some of the argument variables.

2-5.10 Subroutine CON503

COMMON BLOCKS: ZZB

ZZDST

ARGUMENT: LOC

LOC DIR standing location index

Subroutine CON503 processes the CONNECT data name from the COMLINE section. CON503 calls subroutine GETJ with an argument value of one to indicate that a data name is to be retrieved. The retrieved name is inserted into the DIR. If GETJ fails to return a data name, processing of the CONNECT data name ends and subroutine ERMSG is called to produce the following message:

ERROR IN SECTION COMLINE DATA NAME

ERROR NUMBER 16 ERROR LEVEL 100

2-5.11 Function CRND

ARGUMENTS: X,Q

X Variable to be rounded

Q Round off level

ENTRY POINTS: DRND, CINT, DINT

Function CRND rounds or truncates the input value to the specified round off level. Truncation is done by calling the entry points CINT and DINT. Calls to either entry point results in a call to the FORTRAN library function AINT. The DINT entry point is included to maintain compatibility between different computer systems.

Rounding is done by calling the entry points CRND and DRND. Calls to either entry point result in the rounding of the variable according to the expression:

$$CRND = Q * AINT\ (X+Q/2)/Q$$

where CRND, Q, X, and AINT are the rounded result, the round off level, variable to be rounded, and the FORTRAN library function, respectively.

2-5.12 Subroutine ERMSG

COMMON BLOCKS: ZZB

ZZDST

ARGUMENTS: LE, K, LOC

LE	Error level indicator
K	Message identifier
LOC	Array IADS index

Subroutine ERMSG prints messages identifying errors in the input data by indicating the section and data name of the erroneous input data, and the subroutine detecting the error. The messages printed by ERMSG and their corresponding message identifier value are:

K	MESSAGE
1	ALPHABETIC DATA NOT FOUND WHEN EXPECTED
2	MATCH NOT FOUND IN ALPHABETIC TYPE DATA
3	LESS THAN MINIMUM NUMBER OF ITEMS IN NUMERIC LIST
4	INVALID DATA STRUCTURE ITEM IN IADS
5	TOO LATE TO USE THIS DATA NAME
6	IMPLICIT INDEX USED TOO MANY TIMES
7	LIST DATA IN AN ARRAY
8	BLOCK FOUND WITHIN A BLOCK
9	ILLEGAL INPUT TO READDD
10	IDENT NOT FOUND AFTER DEFINE
11	ILLEGAL SECTION - SEE INPUT FORM DOCUMENT

2-5.13 Subroutine FINBLK

COMMON BLOCKS: ZZB

ZZDST

ZZBLK

Subroutine FINBLK completes a DIR or continuation record which contains DIR data block information. The length of the DIR or continuation record is computed and saved before the information is written on the DIR file.

2-5.14 Subroutine GET

COMMON BLOCKS: ZZB
ZZPS
ZZFILA
ZZEDIT

ARGUMENT: K

K Initialization argument

ENTRY POINTS: GET, GETDN, GETIN, GETLM, GETM, GETNM, GETV, GETVS

Subroutine GET retrives the next input item. Subroutine GET has a number of entry points which are used to gain access to specific processing functions of the GET subroutine. The entry points and their associated functions are presented in Table 5. The subroutine uses the two Compass subroutines GETITEM and GETCHAR to perform the input processing of the input buffer.

The initialization argument is always assigned a value of zero when the subroutine is used with the ETC^3 model. The error messages produced by subroutine GET are identified in Section 2-6.

Table 5. Subroutine get entry points.

NAME	ARGUMENTS	FUNCTION
GET	K	Initialize and fill buffers
GETDN	-	Return a data name
GETIN	-	Return name or a character string
GETLM	-	Initialize to fetch a list of numbers
GETM	-	Get a number
GETNM	-	Get the next number in a list of numbers
GETV	-	Get the next input item, a DEFINE is treated as an end-of-data
GETVS	-	Get the next item, this operates like GETV, only DEFINE is returned as a name.

2-5.15 Subroutine GETCHAR

ENTRY POINTS: GETCHR, GETCHN, EXACHR, GETCHY

Subroutine GETCHAR is a Compass subroutine which gets the next character from the input buffer and returns the character to the calling routine. The subroutine interfaces with the other subroutines via registers rather than arguments or common varibles.

There are four entry points to the subroutine for four specific functions. The entry point name and the associated functions are:

(1) GETCHR - Retrieves the next character

(2) GETCHN - Retrieves the next character, but does not return a blank

(3) EXACHR - Retrieves and examines the next character in the input file, but does not advance the input buffer index

(4) GETCHY - Performs miscellaneous functions related to input/output control

2-5.16 Subroutine GETCRD

ARGUMENTS: IEOF, NCARD, FILEC

IEOF	End of file indicator
NCARD	Card image buffer array
FILEC	Logical unit number indicator

Subroutine GETCRD reads a record in card image form from the input file and stores it in a buffer for later processing. It also writes the card image input record to an output file.

2-5.17 Subroutine GETDNC

COMMON BLOCKS: ZZDST

ZZPS

ARGUMENTS: IL, IU, LOC, KLASS, LFK

IL	Lower index of classification data in IADS array
IU	Upper index of classification data in IADS array
LOC	IADS array index of matching data name
KLASS	Data name classification indicator
LFK	Error message print control flag

ENTRY POINT: CHKDNC

Subroutine GETDNC gets a data name from the input buffer and determines its classification. Subroutine GETJ is called to perform the actual retrieval of data from the input buffer.

Entry point CHKDNC is used only to determine the classification of the data name, it bypasses the data name retrieval processing performed by calling subroutine GETJ. In determining the data name classification, incorrect or invalid data names are identified by error messages orginating from this subroutine. The error messages are identified in Section 2-6.

2-5.18 Subroutine GETITEM

COMMON BLOCK: ZZEDIT

Subroutine GETITEM is a Compass subroutine which does the token retrieval from the input buffer. The retrieved tokens are returned via common ZZEDIT. Subroutine GETITEM uses subroutine GETCHAR to perform the character retrieval from the input buffer.

2-5.19 Subroutine GETJ

COMMON BLOCK: ZZPS

ARGUMENT: KYND

KYND Data type indicator

Subroutine GETJ interfaces various subroutines with the GET subroutine. The subroutine calls GETV to locate the next data item which is a data name, numeric quantity, or the DEFINE data name. All other data types are bypassed.

2-5.20 Subroutine GETQ

COMMON BLOCK: ZZPS

ARGUMENT: NRET

NRET Return code from GETQ

Subroutine GETQ processes the input buffer until either a right or left parenthesis or a data name is encountered. Subroutine GETQ calls GETV to read the input buffer.

2-5.21 Subroutine GTDNC2

COMMON BLOCKS: ZZDST

ZZPS

ARGUMENTS: IL, IU, LOC, KLASS, LFK

IL	Lower bound index of classification data in IADS
IU	Upper bound index of classification data in IADS
LOC	IADS index of matching data name
KLASS	Data name classification indicator
LFK	Error message print control flag

Subroutine GTDNC2 will process a data name from the input buffer and determine its classification or validity. Subroutine GETJ is used by GTDNC2 to obtain the data name from the input buffer. The data name is compared with the data names occurring in the IADS array between the specified bounds. The messages which are produced by GTDNC2 are identified in Section 2-6.

2-5.22 Subroutine IEBBLK

COMMON BLOCKS: ZZBLK

ZZDST

ARGUMENT: LOC

LOC IADS array index

Subroutine IEBBLK processes all DIR data blocks in the scenario input data. It calls subroutine ALOBLK to allocate space for the block and subroutine IEBINI to initialize the DIR for the data block.

2-5.23 Subroutine IEBINI

COMMON BLOCKS: ZZB

ZZDST

ARGUMENTS: LB, UB, IBASE, LOFT

LB	Lower bound index of initialization data in IADS
UB	Upper bound index of initialization data in IADS
IBASE	DIR on data block index for initialization
LOFT	Structure level indicator

Subroutine IEBINI initializes DIR's or data blocks from initialization data stored in the IADS array. The specific locations to be initialized and the initialization values for the DIR or data block are contained in the IADS array.

2-5.24 Function IOR

ARGUMENTS: I, J

I	Input variable
J	Input variable

Function IOR performs the logical or operation on the variables I and J.

2-5.25 Function KFIELD

COMMON BLOCKS: Blank

ZZDST

ARGUMENTS: I, K

I IADS array index

K Field indicator

Function KFIELD returns the specification for various fields in the IXTYPE words of the IADS array. The field indicator values used to retrieve the field specifications for the various field types are:

K	FIELD
1	Type
2	Continuation
3	Trigger DIR
4	Use
5	Subroutine

2-5.26 Subroutine LOC

COMMON BLOCKS: ZZB

ZZPS

ARGUMENTS: INDEX

NRET

INDEX Index to LOCATION data name data in DIR

NRET Return code indicator

Subroutine LOC processes the data name LOCATION or its abbreviated form LOC. The subroutine processes the latitude, longitude, altitude and the UTM location data of a position specified by the scenario input data. The subroutine is called after the data name LOCATION or LOC is encountered, and expects the location data to follow in the order of latitude, longitude, and altitude, or UTM coordinates and altitude. The altitude can be either mean sea level or height above terrain. The error messages produced by subroutine LOC are identified in Section 2-6.

2-5.27 Subroutine LOCFG

COMMON BLOCKS: ZZB

ZZPS

ARGUMENT: SUM

SUM Radian equivalent of angular measures

ENTRY POINTS: LOCF, LOCG

Subroutine LOCFG processes latitudes and longitudes by converting the degrees, minutes, seconds into radians of arc. The subroutine checks the input latitudes and longitudes to determine that the values are within the acceptable maximum and minimum limits. Error messages produced by the subroutine are identified in Section 2-6.

The main entry point is not used, but the entry points LOCF and LOCG are used for latitude and longitude inputs, respectively. The two entry points are used to establish the maximum and minimum limit values used in checking the input. The degrees, minutes, and seconds to radians conversion computations are the same for latitude and longitude values.

2-5.28 Subroutine LOCTN

ARGUMENT: INDEX

INDEX Subroutine LOC argument

Subroutine LOCTN is an interface subroutine. Other subroutines requiring the processing done by LOC, but not needing the return code provided by LOC, call subroutine LOCTN. Subroutine LOCTN calls LOC with a dummy parameter for the return parameter.

2-5.29 Subroutine MSD

ARGUMENTS: DEST, SOUR, NCHAR

DEST	Transferred characters' destination array
SOUR	Transferred characters' source array
NCHAR	Number of transferred characters

Subroutine MSD transfers characters from one location to another location in groups of four characters. The remaining character positions are filled with period marks. Characters are transferred in groups of four until all characters are transferred.

2-5.30 Subroutine MSTRSB

COMMON BLOCKS: ZZB
ZZDST
ZZPS
ZZBLK

ARGUMENTS: IPARM, LOC

IPARM	Subroutine indicator
LOC	Index of calling entry in IADS array

Subroutine MSTRSB calls other subroutines according to the value of the subroutine indicator parameter. It is retrieved from the IADS array. The subroutine number, JJ, is retrieved from the IADS array via the subroutine indicator parameter. The subroutine number and the processing function or subroutine performing the function are indicated below.

JJ	Processing Function or Subroutine
1	Place DIR data block on continuation record
2	Use negative absolute value of numeric parameter
501	Process CONTROL section with subroutine PCONT
503	Process CONNECT data name with subroutine CON503
505	Process THREATBM section with subroutine PATHVB
507	Process error messages with subroutine ERMSG
508	Process VULNUM data name with subroutine VN508

2-5.31 Subroutine PAGE

ARGUMENTS: KK, NUNIT

KK	Print line counter
NUNIT	Logical unit number indicator

ENTRY POINTS: PAGEA, PAGEB, PAGEC, PAGED

Subroutine PAGE controls the classification marking and pagination of printed output. The specific classification marking and pagination functions are entered through the alternate entry points. The entry points and their processing function are described below:

(1)	PAGE	Controls pagination
(2)	PAGEA	Receives Classification marking
(3)	PAGEB	Closes output for NUNIT
(4)	PAGEC	List data card
(5)	PAGED	Set up messages for specified unit

2-5.32 Subroutine PCONT

COMMON BLOCKS: ZZB

ZZPS

Subroutine PCONT processes the CONTROL section of the scenario input data. The subroutine reads and processes all the data elements appearing in this section of the input data. The subroutine processes the identifying data from the CONTROL section and inserts the information into the structured DIR data base for identification purposes. It also checks for incorrect and invalid entries and produces error or informative messages identifying the input. Error messages from this subroutine are identified in Section 2-6.

2-5.33 Subroutine PLL

COMMON BLOCKS: ZZB

ZZPS

ARGUMENTS: UNAME, INDEX

UNAME Array of names used in error messages

INDEX UNAME array index

ENTRY POINTS: PLAT, PLONG

Subroutine PLL is used to process input latitudes and longitudes. The subroutine is entered via either the PLAT or PLONG entry points, but not the main entry point PLL. The processing performed by either entry point is identical except for the check of the input value against the minimum and maximum limit values.

Processing consists of interpreting the latitude and longitude values in the scenario input data, determining their validity, and converting the given latitudes and longitudes into radians. Detected input errors are identified by appropriate names in the error messages. Section 2-6 summarizes the errors detected by PLL and the error messages produced.

2-5.34 Subroutine PTHVB

COMMON BLOCKS: ZZB

ZZCVX

ZZPS

Subroutine PTHVB processes ballistic missile threat data from the THREATBM section of the scenario input data. The subroutine processes the input data for errors in data names and initializes variables which describe threats. It also assigns default values for data elements which were not provided input values. The error messages produced by subroutine PATHVB are summarized in Section 2-6.

2-5.35 Subroutine READDD

COMMON BLOCKS: ZZB

ZZDST

ARGUMENT: IUNIT, D

IUNIT Device number of specification input file

D D is the device number of the output file

Subroutine READDD reads the data definitions and DIR specifications from the specification file. The subroutine constructs the table which provides the necessary information to transform the scenario input data into DIRs. One section of READDD processes the data definitions and another processes the DIR specifications.

2-5.36 Subroutine SCAN

COMMON BLOCKS: Blank

ZZPS

ZZDST

ARGUMENTS: LL, LU, LOC

LL	Lower bound search index for IADS array
LU	Upper bound search index for IADS array
LOC	IADS index of matching data name

Subroutine SCAN searches for a match of the current data name in the IADS array between the supplied bounds.

2-5.37 Subroutine UNITCH

COMMON BLOCKS: ZZB

ZZPS

ZZCVX

ENTRY POINT: UNITST

Subroutine UNITCH establishes a system of units to be used in processing input data. Table 6 identifies the systems currently used. Systems of units not corresponding to those specified in the table are processed unchanged. The entry point UNITST results in the usage of the Standard system of units in Table 6. The conversion factors for these systems of units are initialized and stored by subroutine UNITCH.

Table 6. Systems of units.

GROUP	STANDARD	JSIPSA	JSIPSB
RANGE	kilometers	nautical miles	nautical miles
LENGTH	meters	nautical miles	nautical miles
HEIGHT (ALTITUDE)	meters	kilofeet	nautical miles
TIME	seconds	seconds	seconds
VELØCITY	meters/sec	knots	feet/sec
ANGLE	degrees	degrees	degrees
PRØBABILITY	0.0-1.0	0.0-1.0	0.0-1.0
YIELD	kilotons	kilotons	kilotons
FREQUENCY	MHz	MHz	MHz
ØVERPRESSURE	psi	psi	psi
FUEL QUANTITY	gallons	gallons	gallons
FUEL RATE	gallons/sec	gallons/sec	gallons/sec
ANGULAR VELØCITY	degrees/sec	degrees/sec	degrees/sec

2-5.38 Subroutine UTMGC

ARGUMENTS: AE, AN, AA, AB, AF, AG

AE	Easting, central meridian easting to 500,000
AN	Northing
AA	Semi-major axis of projection spheriod
AB	Semi-minor axis of projection spheriod
AF	On input, hemisphere indicator On output, latitude
AG	On input, central meridian On output, longitude

Subroutine UTMGC performs the UTM to geodetic coordinate conversion. The basic reference used in the derivation of the calculations is TM 5-241-8, dated April 1973. Subroutine UTMGC uses the projection spheroid determined by UTMSP and the easting and northing computed by subroutines UTME and UTMN respectively to do the UTM to geodetic coordinate conversion.

2-5.39 Subroutine UTME

ARGUMENTS: NZ, NHL, BE, SE

NZ	UTM zone indicator
NHL	UTM 100 Km square designator
SE	Easting of zone specified by NZ
BE	Easting from the UTM zone central meridian

Subroutine UTME computes the easting from a given zone. The total easting is computed by determining the number of zones intervening and multiplying by 100,000. The easting within the zone is then added to obtain the total easting.

2-5.40 Subroutine UTMGEO

ARGUMENTS:	NZONE, LTRS, EAST, XNTH, XLAT, XLONG, NER
NZONE	UTM zone designator
LTRS	UTM 100 Km square designator
EAST	Easting
XNTH	Northing
XLAT	Latitude
XLONG	Longitude
NER	Error level code

Subroutine UTMGEO converts UTM locations to geodetic coordinates. It is a driver program which calls other subroutines to perform specific computations to do portions of the coordinate conversion. Subroutine UTMGEO performs error checking on the input data and produces error and informative messages identifying incorrect or invalid inputs. The messages produced by this subroutine are identified in Section 2-6.

After subroutine UTMGEO completes error checking, subroutine UTMSP is called to determine the proper spheriod projection to be used. Subroutines UTME and UTMN are called to compute the eastings and northings respectively, and UTMGC is called to perform the actual coordinate conversion.

2-5.41 Subroutine UTMN

ARGUMENTS: NZ, NZL, NSP, NHL, SN, BN

NZ	UTM zone indicator
NZL	UTM zone letter
NSP	UTM projection spheriod
NHL	UTM 100 Km grid designator
SN	UTM northing following 100 Km grid designator
BN	Northing from the equator

Subroutine UTMN computes the northing of a given UTM zone and zone letter. If the UTM zone and zone letter indicate that the location is in the southern hemisphere a false northing is added.

2-5.42 Subroutine UTMSP

ARGUMENTS: NZ, NZL, NSP, AA, AB

NZ	UTM zone indicator
NZL	UTM zone letter
NSP	UTM projection spheriod
AA	Semi-major axis of projection spheroid
AB	Semi-minor axis of projection spheroid

Subroutine UTMSP determines which one of the standardized projection spheriods is to be used in the UTM to geodetic coordinate conversion calculations. The subroutine specifies the semi-major and minor axes to be used in the computations. These values are determined by using the UTM zone indicator as an index into a table. The table is a cross reference of UTM zone and spheroid identifiers. The UTM zone and zone letter information identify the proper spheroid to be used in the coordinate conversion.

2-5.43 Subroutine VN508

COMMON BLOCKS: ZZB

ZZDST

ZZPS

ARGUMENT: LOC

LOC Index into IADS for vulnerability number information

Subroutine VN508 is called when the Input Editor encounters VULNUMBERKILL or VNK. It calls subroutine GETJ to retrieve the next two data entries. The first must be numeric and have a value between .9 and 100. If the value does not fall in this range, ERMSG is called and VN508 continues with processing. The second entry must be two characters long, and if the entry is not valid, ERMSG is called before control is returned from VN508. A valid value for VULNUMBERKILL results in it being stored in the DIR.

2-5.44 Subroutine WDIR

COMMON BLOCKS: ZZB
ZZBLK

Subroutine WDIR writes a DIR, a DIR and a continuation record, or a continuation record before starting another DIR. Error checking is done on the DIR and the variable MDLEN. The value of MDLEN must be greater than zero. The DIR or continuation record is completed by subroutine FINBLK and is written onto TAPE7 and TAPE13 by subroutine WRCD. The DIR indicator is assigned a logical value of FALSE after the DIR is written or if any errors are detected. Control is returned to the calling program after all of the above has been performed.

2-5.45 Subroutine WRCD

ARGUMENTS: FMT, TYP, CNT, LEN, BUFFER

FMT	DIR format
TYP	DIR type
CNT	DIR continuation record flag
LEN	DIR length
BUFFER	DIR storage area

Subroutine WRCD writes the format, type, continuation and length information about a DIR as well as the DIR itself in unformatted binary form on TAPE7. The subroutine also produces a printable copy of the DIR on TAPE13. The descriptive data such as format, type, continuation, and length information are written as a heading and each field is unpacked and printed in the appropriate format.

2-6 REFERENCE TABLES-IEDITB

Three tables are presented to indicate to the programmer the way in which the IEDITB subprogram units are interrelated with respect to each other and to the common variables. In addition, a table is included to identify the entry points of a subprogram unit, and a table is included to identify the error messages produced by the IEDITB program and the subprogram unit responsible for producing the message.

2-6.1 IEDITB Subroutine Entry Point Table

Table 7 identifies the alternate entry points for each subprogram unit of the IEDITB program.

2-6.2 IEDITB Subroutine Call Table

Table 8 identifies the subprogram units called, if any, by each subprogram unit of the IEDITB program.

2-6.3 IEDITB Call Structure Table

Table 9 identifies the hierarchical sequency of subprogram calls made by the IEDITB program. The names appearing in parentheses are entry points into subprogram units. Table 8 provides the means of identifying the subprogram unit name.

2-6.4 IEDITB Common Update Table

Table 10 identifies the variables in the various common blocks which are used or updated by the subprogram units of IEDITB. A variable in common is used if its value is referred to by any executable statement, and is updated if it is assigned a value by any DATA or executable statement.

2-6.5 IEDITB Error Messages

Table 11 lists the error messages which are produced by the IEDITB subprogram units. These error messages are produced as an aid to the user and programmer in running the IEDITB program. Table 12 also identifies the subroutine which will produce the message.

Table 7. IEDITB subroutine entry points.

SUBROUTINE	ENTRY POINTS						
GETITEM	GETITM						
GETCHAR	GETCHR	GETCHN	EXACHR	GETCHY			
CKD	CKP						
PLL	PLAT	PLONG					
GET	GETDIN	GETIN	GETLM	GETM	GETNM	GETV	GETVS
UNITCH	UNITST						
GRND	CINT	DINT	DRND				
ARHLOC	LOCARH						
GETDNC	CHKDNC						
PAGE	PAGEA	PAGEB	PAGEC	PAGED			
LOCFG	LOCF	LOCG					

Table 8. IEDITB subroutine call. (1 of 3)

CALLED \ CALLING	ALOBLK	ARHLOC	AUTOPB	AZRNGE	BCDB	BOUNDS	CK	CKD	CON503	CRND	EBCDZ	EQC	ERMSG	FINBLK	GEDUTM	GET	GETCHAR	GETDNC	GETITEM	GETJ	GETQ
1. ALOBLK																					
2. ARHLOC				X																	
3. AUTOPB																					
4. AZRNGE																					
5. BCDB																					
6. BOUNDS			X																		
7. CK																					
8. CKD																					
9. CON503																					
10. CRND							X	X													
11. EBCDZ																					
12. EQC																					
13. ERMSG			X						X												
14. FINBLK	X																				
15. GEOUTM																					
16. GET			X																	X	X
17. GETCHAR																X			X		
18. GETDNC			X																		
19. GETITEM																X					
20. GETJ			X						X									X			
21. GETQ																					
22. GTDNC2																					
23. HF502																					
24. IEBAIR			X																		
25. IEBARR			X																		
26. IEBBLK			X																		
27. IEBINI			X																		
28. IEDITB																					
29. IOR																					
30. KFIELD			X			X												X			
31. LOC																					
32. LOC504																					
33. LOCFG																					
34. LOCTN																					
35. MSTRB																		X			
36. MVC									X												
37. PAGE							X	X					X			X		X			
38. PCONT																					
39. PLL																					
40. PRC509																					
41. PTHVA																					
42. PTHVB																					
43. READDD																					
44. SCAN			X															X			
45. TIMDAT																					
46. VN508																					
47. UNITCH																					
48. UTME																					
49. UTMGC																					
50. UTMGEO																					
51. UTMN																					
52. UTMSP																					
53. WDIR			X																		
54. WRCD	X																				
55. ZBOOM																			X		

Table 8. IEDITB subroutine call. (2 of 3)

CALLED \ CALLING	GTONC2	HF502	IEBAIR	IEBARR	IEBBLK	IEBINI	IEDITB	IOR	KFIELD	LOC	LOC504	LOCFG	LOCTN	MSTRB	MVC	PAGE	PCONT	PLL	PRC509	PTHVA	PTHVB
1. ALOBLK					X																
2. ARHLOC																					
3. AUTOPB							X														
4. AZRNGE																				X	
5. BCDB										X											
6. BOUNDS		X	X	X	X						X										
7. CK			X	X																X	X
8. CKD																		X		X	X
9. CON503																			X		
10. CRND																					
11. EBCDZ										X											
12. EQC										X											
13. ERMSG		X	X	X	X						X			X							
14. FINBLK																					
15. GEOUTM										X											
16. GET							X			X		X					X	X		X	X
17. GETCHAR																X					
18. GETDNC					X																
19. GETITEM																					
20. GETJ	X		X	X							X										
21. GETQ		X																			
22. GTDNC2											X										
23. HF502														X							
24. IEBAIR					X						X										
25. IEBARR					X																
26. IEBBLK																					
27. IEBINI					X						X										
28. IEDITB																					
29. IOR			X																		
30. KFIELD			X	X	X	X															
31. LOC													X							X	
32. LOC504														X							
33. LOCFG										X											
34. LOCTN			X								X										
35. MSTRB																					
36. MVC		X	X			X				X						X					
37. PAGE	X						X			X		X									
38. PCONT														X							
39. PLL																				X	X
40. PRC509														X							
41. PTHVA														X							
42. PTHVB														X							
43. READDD							X														
44. SCAN	X	X	X	X																	
45. TIMDAT																					
46. VN508																					
47. UNITCH							X														
48. UTME																					
49. UTMGC																					
50. UTMGEO										X											
51. UTMN																					
52. UTMSP																					
53. WDIR																					
54. WRCD																	X				
55. ZBOOM																					

Table 8. IEDITB subroutine call. (3 of 3)

CALLED \ CALLING	READDD	SCAN		TIMDAT	VN508		UNITCh	UTME		UTMGC	UTMGEO		UTMN	UTMSP		WDIR	WRCD		ZBOOM		
1. ALOBLK																					
2. ARHLOC																					
3. AUTOPB																					
4. AZRNGE																					
5. BCDB											X										
6. BOUNDS																					
7. CK																					
8. CKD																					
9. CON503																					
10. CRND																					
11. EBCDZ																					
12. EQC	X	X			X	X															
13. ERMSG	X				X																
14. FINBLK																X					
15. GEOUTM																					
16. GET																					
17. GETCHAR																					
18. GETDNC																					
19. GETITEM																					
20. GETJ					X		X														
21. GETQ																					
22. GTDNC2																					
23. HF502																					
24. IEBAIR																					
25. IEBARR																					
26. IEBBLK																					
27. IEBINI																					
28. IEDITB																					
29. IOR																					
30. KFIELD		X																			
31. LOC																					
32. LOC504																					
33. LOCFG																					
34. LOCTN																					
35. MSTRB																					
36. MVC											X			X			X				
37. PAGE	X						X														
38. PCONT																					
39. PLL																					
40. PRC509																					
41. PTHVA																					
42. PTHVB																					
43. READDD																					
44. SCAN																					
45. TIMDAT																					
46. VN508																					
47. UNITCH																					
48. UTME											X										
49. UTMGC											X										
50. UTMGEO																					
51. UTMN											X										
52. UTMSP											X										
53. WDIR																					
54. WRCD																X					
55. ZBOOM																					

Table 9. IEDITB call structure table. (1 of 4)

LINE NO \ COLUMN	0	1	2	3	4	5	6	7	8	9	10
1	IEDITB	AUTOPB	BOUNDS	KFIELD							
2			(CHKDNC)	GETJ	(GETV)	(GETCHY)					
3						(GETITM)	(EXACHR)				
4							(GETCHR)				
5							(GETCHY)				
6							(GETCHN)				
7							ZBOOM				
8						PAGE	(GETCHY)				
9							MVC				
10				KFIELD							
11				MSTRB	CON503	ERMSG	PAGE(8,5)				
12						GETJ(2,3)					
13						MVC					
14					ERMSG	PAGE					
15					HF502	BOUNDS	KFIELD				
16						ERMSG	PAGE(8,5)				
17						GETQ	(GETV)(2,4)				
18						MVC					
19						SCAN	EQC				
20							KFIELD				
21					LOC504	BOUNDS	KFIELD				
22						ERMSG	PAGE(8,5)				
23						GETJ(2,3)					
24						GTDNC2	GETJ(2,3)				
25							PAGE(8,5)				
26							SCAN(19,5)				
27						IEBAIR	BOUNDS	KFIELD			
28							CK	CRND			
29								PAGE(8,5)			
30							ERMSG	PAGE(8,5)			
31							GETJ(2,3)				
32							IOR				

Table 9. IEDITB call structure table. (2 of 4)

COLUMN / LINE NO	0	1	2	3	4	5	6	7	8	9	10
33							KFIELD				
34							LOCTN	LOC	BCDB		
35									EBCDZ		
36									EQC		
37									GEOUTM		
38									(GETV)(2,4)		
39									(LOCF)	(GETV)(2,4)	
40										PAGE(8,5)	
41									(LOCG)(39,8)		
42									MVC		
43									PAGE(8,5)		
44									UTMGEO	BCDB	
45										EQC	
46										MVC	
47										UTME	
48										UTMGC	
49										UTMN	
50										UTMSP	MVC
51									MVC		
52									SCAN(19,5)		
53						IEBINI	KFIELD				
54							MVC				
55						LOCTN					
56					PCONT	(GETDN)(2,4)					
57						(GETIN)(2,4)					
58					(GETV)(2,4)						
59					PAGE(8,5)						
60					(PAGEA)(8,5)						
61					TIMDAT						
62					WRCD	MVC					
63						PAGE(8,5)					
64					PRC509						
65					PTHVA	AZRNGE	(LOCARH)				
66						CK(28,6)					

Table 9. IEDITB call structure table.(3 of 4)

COLUMN / LINE NO	0	1	2	3	4	5	6	7	8	9	10
67						CKD	CRND				
68							(DRND)				
69							PAGE				
70						(GETDN)(2,4)					
71						(GETIN)(2,4)					
72						(GETLM)(2,4)					
73						(GETM)(2,4)					
74						(GETNM)(2,4)					
75						LOC(34,7)					
76						PAGE(8,5)					
77						(PLAT)	CKD(67,5)				
78							(GETM)(2,4)				
79							(GETV)(2,4)				
80							PAGE(8,5)				
81						(PLONG)(77,5)					
82						UNITCH	EQC				
83							GETJ(2,3)				
84							PAGE(8,5)				
85					PTHVB	CK(28,6)					
86						CKD(67,5)					
87						(GETDN)(2,4)					
88						(GETIN)(2,4)					
89						(GETM)(2,4)					
90						PAGE(8,5)					
91						(PLAT)(77,5)					
92						(PLONG)(77,5)					
93						UNITCH(82,5)					
94					VN508	EQC					
95						ERMSG	PAGE(8,5)				
96						GETJ(2,3)					
97				PAGE(8,5)							
98				SCAN(19,5)							
99			ERMSG	PAGE(8,5)							
100			GETDNC(2,2)								

Table 9. IEDITB call structure table. (4 of 4)

COLUMN / LINE NO	0	1	2	3	4	5	6	7	8	9	10
101			GETJ(2,3)								
102			(GETVS)(2,4)								
103			IEBAIR(27,5)								
104			IEBARR	BOUNDS	KFIELD						
105				CK(28,6)							
106				ERMSG	PAGE(8,5)						
107				GETJ							
108				KFIELD							
109				SCAN(19,5)							
110			IEBBLK	ALOBLK	FINBLK						
111					WRCD(62,5)						
112				BOUNDS	KFIELD						
113				(CHKDNC)(2,2)							
114				ERMSG	PAGE(8,5)						
115				GETDNC(2,2)							
116				IEBAIR(27,5)							
117				IEBARR(104,2)							
118				IEBINI(53,5)							
119				KFIELD							
120			IEBINI(53,5)								
121			KFIELD								
122			SCAN(19,5)								
123			WDIR	FINBLK							
124				WRCD(62,5)							
125		GET(2,4)									
126		PAGE(8,5)									
127		(PAGEC)(8,5)									
128		READDD	EQC								
129			ERMSG	PAGE(8,5)							
130			PAGE(8,5)								
131		(UNITST)									

AD-A077 617 BDM CORP MCLEAN VA F/G 17/2
EUROPEAN THEATER COMMAND, CONTROL AND COMMUNICATIONS (ETC3) MOD--ETC(U)
NOV 78 P BOGUE , R LABASH , J SANDERS DNA001-78-C-0070
UNCLASSIFIED BDM/W-78-718-TR DNA-4351T-2B NL

2 OF 4
ADA
077617

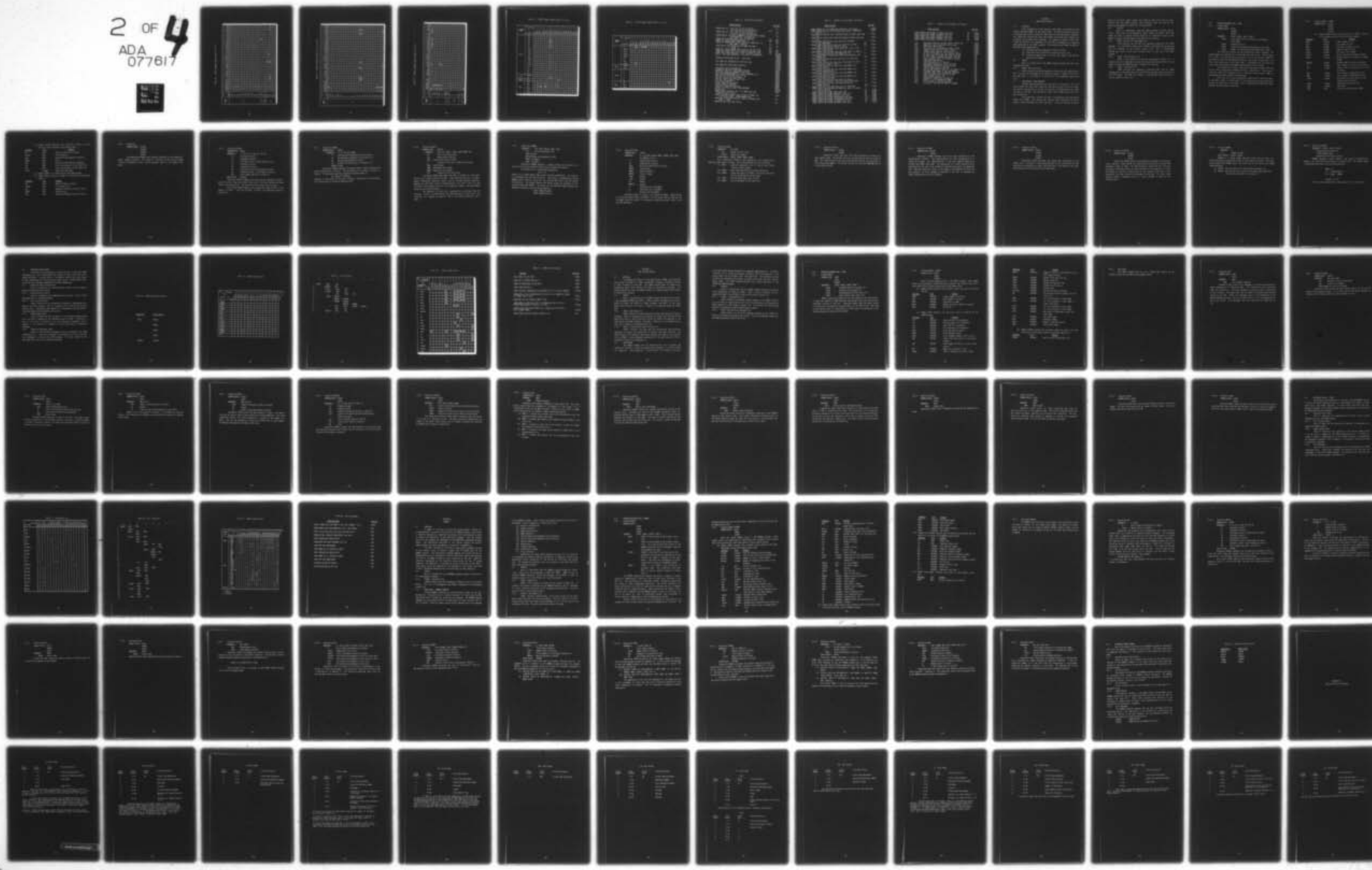


Table 10. IEDITB common update table. (1 of 5)

COMMON BLOCK	VARIABLE \ ROUTINE	ALOBLK	ARHLOC	AUTOPB	AZRNGE	BCDB	BOUNDS	CK	CKD	CON503	CRND	EBCDZ	EOC	ERMSG	FINBLK	GEOUTM	GET	GETCHAR	GETDNC	GETITEM	GETJ	GETO	GTDNC2	HF502	IEBAIR	IEBARR	IEBBLK	IEBINI	IEDITB	IOR	KFIELD	LOC	LOC504	LOCFG	LOCTN	MSTRB
ZZDST	IBDD																																			
	ILDD																																			
	IODD																																			
	IBSS			○																																
	ILSS			○			○																		○	○	○	○								
	IOSS			○			○																													
	IXREF																																			
	IXUNIT																								○	○										
	IXSCAL																																			
	IXMIN																								○	○										
	IXMAX																								○	○										
	IXQNTM																								○	○										
	IXTYPE			○			○												○						○	○	○	○								
	IXNAME													○										○	○	○		○								
	IXDA			○						○				○										○	○	○		○								
	IXDB			○																				○	○	○		○								
	IXDC			○						○															○			○								
	IXDD			○																					○			○								
BLANK	IADMAX																												●							
	ADS			○						○				○										○	○	○	○	○			○					
ZZCVX	CVX							○	○																											
ZZPS	LEVEL																●							○								○				
	KIND			○						○							●		○		●	○	○		○	○						○		○		
	CBUF			○				●	●	○							●		○				○		○							○		●		●
	KIND9																																			
	KIND2																●								○											○
	CBUFZQ																																			

Table 10. IEDITB common update table. (2 of 5)

COMMON BLOCK	VARIABLE \ ROUTINE	ALOBLK	ARHLOC	AUTOPB	AZRNGE	BCDB	BOUNDS	CK	CKD	CON503	CRND	EBCD2	EOC	ERMSG	FINBLK	GEOUTM	GET	GETCHAR	GETDNC	GETITEM	GETJ	GETQ	GTDNCZ	HF502	IEBAIR	IEBARR	IEBBLK	IEBINI	IEDITB	IOR	KFIELD	LOC	LOC504	LOCFG	LOCTN	MSTRB
ZZB	SECNAM			●				○	○					○																						
	PACKRD																																			
	PACKR																																			
	TMAX																																			
	TQUANT																																			
	DBLANK																							○												
	DZERO																																			
	D2R																																	○		
	R2D																																			
	R15																																			
	R30																																			
	R45																																			
	R60																																			
	R90																																			
	R180																																			
	R360																																			
	LERR							●	●					●			●															●		●		
	NERR							●	●					●			●															●		●		
	LERRP																																			
	LERRS																																			
	MSCNT																																			
	NVEHA																																			
	NVEHB																																			
	NERRA																																			
	NERRB																																			
	MDFMT	○		●																																
	MDTYP	○		●																																
	MDCNT	●		●																																
	MDLEN	○		●											●																		●			

Table 10. IEDITB common update table. (3 of 5)

COMMON BLOCK	VARIABLE \ ROUTINE	ALOBLK	ARHLOC	AUTOPB	AZRNGE	BCDB	BOUNDS	CK	CKD	CON503	CRND	EBCDZ	EQC	ERMSG	FINBLK	GEOUTM	GET	GETCHAR	GETDNC	GETITEM	GETJ	GETQ	GTDNC2	HF502	IEBAIR	IEBARR	IEBBLK	IEBINI	IEDITB	IOR	KFIELD	LOC	LOC504	LOCFG	LOCTN	MSTRB
ZZB	CD	●								○					●									○	●	●		●				●	●			
	MD2FMT	○		●																																
	MD2TYP	○		●																																
	MD2CNT	●		●																																○
	MD2LEN	○		●											●																					
	CD2	●		●											●																					
	ND12	○																																		
	LZONE																															○				
ZZFILA	DDNAME																																			
	KFILA																																			
ZZUNIT	U1		○		○																															
	U2		○																																	
	U3																																			
	U4				○																															
	U5																																			
ZZBLK	DIR																																			
	BLKPT																																			
	KODE	●																																		●

Table 10. IEDITB common update table. (4 of 5)

COMMON BLOCK	VARIABLE \ ROUTINE	MVC	PAGE	PCONT	PLL	PRC509	PTHVA	PTHVB	READDD	SCAN	TIMDAT	VN508	UNITCH	UTME	UTMGC	UTMGEO	UTMN	UTMSP	WDIR	WRCD	ZBOOM
ZZDST	IBDD								○												
	ILDD								○												
	IQDD								●												
	IBSS								●												
	ILSS								○	○											
	IQSS								●												
	IXREF																				
	IXUNIT																				
	IXSCAL								○												
	IXMIN								○												
	IXMAX								○												
	IXQNTM								○												
	IXTYPE								○	○											
	IXNAME								○	○											
	IXDA								○			○									
	IXDB								○												
	IXDC																				
	IXDD								○												
BLANK	IADMAX																				
	ADS								●	○		○									
ZZCVX	CVX							○					●								
ZZPS	LEVEL				○																
	KIND			○			○	○				○	○								
	CBUF			○	●		●	○		○		○	○								
	KIND9																				
	KIND2				○																
	CBUF2Q																				

Table 10. IEDITB common update table. (5 of 5)

COMMON BLOCK	VARIABLE \ ROUTINE	MVC	PAGE	PCONT	PLL	PRC509	PTHVA	PTHVB	READDD	SCAN	TIMDAT	VN508	UNITCH	UTME	UTMGC	UTMGEO	UTMN	WDIR	WRCD	ZBOOM
	CD			●			●	●				○						○		
	MD2FMT																	○		
	MD2TYP																	○		
	MD2CNT																	○		
	MD2LEN																	○		
	CD2																	○		
	ND12																			
	LZONE			●																
ZZFILA	DDNAME																			
	KFILA																			
ZZUNIT	U1																			
	U2																			
	U3																			
	U4																			
	U5																			
ZZBLK	DIR																	●		
	BLKPT																			
	KODE																			

Table 11. IEDITB error messages.

ERROR MESSAGE		ROUTINE
ERROR aaaa aa = nnnn.nnnn IS OUT OF LIMITS nnnn.nnnn WILL BE USED INSTEAD	nnn	CK
ERROR aaaa aa = nnnn.nnnn IS OUT OF LIMITS nnnn.nnnn WILL BE USED INSTEAD		CKD
ERROR-PROCESS LAT/LONG A NUMBER(S) IS(ARE) OUT OF RANGE DATA-NAME=aaaa SECTION=aaaa		PLL
ERROR GETDN NO NAME FOUND, nnn,nnn,0000,aaa	100	GET
ERROR GET SYNTAX ERROR-PARENTHESES	100	GET
ERROR GET NUMBERS/DATA FOUND INSTEAD OF NAME-SKIP, TO NEXT NAME. KIND=nn	100	GET
ERROR GETIN CHAR STRING USED AS NAME TOO LONG, 16 USED	100	GET
ERROR BET SYNTAX ERROR, EQUAL SIGN (=) OUT OF PLACE	100	GET
ERROR GET SYNTAX ERROR-LEFT PAREN HAS NO EQUAL SIGN	100	GET
ERROR UNITS NAME aaaa aaaa IS NOT A SYSTEM OF UNITS	AO	UNITCH
EGAD -- ADS TABLE OVERFLOW -- NO POINT IN CONTINUING --		READDD GTDNCZ
UNEXPECTED DATA-NAME-TO NIT - aaaa aaaa		GTODNC GTDNCZ
DATA NAME NOT RECOGNIZED-TO WIT-aaaa aaaa		GETDNC GTDNCZ
DATA NAME NOT FOUND WHERE EXPECTED		GETDNC GTDNCZ
UNEXPECTED END-OF-INFORMATION		GETDNC
ALPHABETIC DATA NOT FOUND WHEN EXPECTED		ERMSG
MATCH NOT FOUND IN ALPHABETIC TYPE DATA		ERMSG
LESS THAN MINIMUM NUMBER OF ITEMS IN NUMERIC LIST		ERMSG
INVALID DATA STRUCTURE ITEM IN IADS		ERMSG
TOO LATE TO USE THIS DATA NAME		ERMSG
IMPLICIT INDEX USED TOO MANY TIMES		ERMSG
LIST DATA IN AN ARRAY		ERMSG
BLOCK FOUND WITHIN A BLOCK		ERMSG
ILLEGAL INPUT TO READDD		ERMSG
IDENT NOT FOUND AFTER DEFINE		ERMSG
ILLEGAL SECTION-SEE INPUT FORM DOCUMENT		ERMSG
END-OF-FILE		GETCRD
ERROR PROCESS CONTROL-TYPE = nnn NBUF=aaaa aaaa KIND=nnnn (See Table)		PCONT
ERROR LOCATION LAT/LONG VALUE IS OUT OF RANGE, OR HAS SYNTAS ERROR. ERROR NUMBER=nnnn		LOCFG
ERROR LOC NO 9109 UTM ODD NO. DIGITS, UTM=aaaa aaaa NBUF1 = 00000		LOC
ERROR LOC NO. nnnn (See Table)		LOC

Table 11. IEDITB error messages (continued).

ERROR MESSAGE		ROUTINE
ERROR UTMGEO NO. nnnn ZONE=aaaa LTRS=aaaa (See Table)		UTMGEO
ERROR THREAT VEHICLE-AST DATA NAME aaaa aaaa IS UNDEFINED 0000 90		PTHVA
ERROR THREAT VEHICLES ASST TIME DUPLICATED IN TRACK aaaa aaaa 100		PTHVA
ERROR THREAT VEHICLES-AST EARP DUPLICATED IN TRACK aaaa aaaa 100		PTHVA
ERROR THREAT VEHICLES-AST aaaa aaaa HAS TWO DROPS AT THE SAME POINT	50	PTHVA
ERROR THREAT VEHICLE-AST aaaa aaaa VELOCITY TO POINT IS NOT GIVEN DEFAULT VALUE IS 200.0	nnn 71	PTHVA
ERROR THREAT VEHICLES-ASST aaaa aaaa POINT nnnn IS TOO CLOSE TO NEXT POINT	91	PTHVA
ERROR THREAT VEHICLES-AST aaaa aaaa TIME INPUT IS INVALID THE VEHICLE MUST NOT START BEFORE T=0.0	91	PTHVA
FATAL ERROR THREAT VEHICLE-AST TABLE aaaa OVERFLOW. RUN TERMINATED	900	PTHVA
ERROR THREAT VEHICLE-AST aaaa aaaa HAS LESS THAN TWO TRACK POINTS	101	PTHVA
ERROR THREAT VEHICLE-AST aaaa aaaa IS MISSING FOR POINT nnn DEFAULT VALUE = nnnn.n	60	PTHVA
ERROR THREAT VEHICLES-AST aaaa aaaa ASM NUMBER nnnn HAS NO VELOCITY	72	PTHVA
ERROR THREAT VEHICLES-AST aaaa aaaa ASM NUMBER nnnn HAS NO CHARACTERISTICS	100	PTHVA
ERROR THREAT VEHICLES-AST aaaa aaaa ASM NUMBER nnnn HAS NO BURST LOCATION	60	PTHVA
ERROR THREAT VEHICLES-AST aaaa aaaa ASM NUMBER nnnn HAS NO BURST ALTITUDE	30	PTHVA
ERROR THREAT VEHICLES-AST aaaa aaaa THE aaaa IS DUPLICATED OR DOESN'T FOLLOW THE ASM IDENTIFIERS	63	PTHVA
ERROR THREAT VEHICLES-AST aaaa aaaa HAS NO CELL CHARACTERISTICS	100	PTHVA
ERROR THREAT-BAT DATA NAME aaaa aaaa IS UNDEFINED	90	PTHVB
ERROR THREAT VEHICLES-BM DATA NAME aaaa aaaa IS ILLEGAL IN BM DEFINITION	90	PTHVB
ERROR THREATS-BM ASSUME IDENT aaaa aaaa	50	PTHVB
ERROR THREATS-BM ASSUME NUMOBJECTS aaaa aaaa	20	PTHVB
ERROR THREATS-BM ASSUME VELOCITY aaaa aaaa	40	PTHVB
ERROR THREATS-BM ASSUME LAUNCHANGLE aaaa aaaa	20	PTHVB
ERROR THREATS-BM ASSUME IDTHREATCHAR aaaa aaaa	100	PTHVB
ERROR THREATS-BM ASSUME LATLAUNCH aaaa aaaa	40	PTHVB
ERROR THREATS-BM ASSUME LONGLAUNCH aaaa aaaa	40	PTHVB

Table 11. IEDITB error messages (continued).

	ERROR MESSAGE		ROUTINE
ERROR	THREATS-BM ASSUME ALTLAUNCH aaaa aaaa	20	PTHVB
ERROR	THREATS-BM ASSUME TIMLAUNCH aaaa aaaa	20	PTHVB
ERROR	THREATS-BM ASSUME LATBURST aaaa aaaa	60	PTHVB
ERROR	THREATS-BM ASSUME LONGBURST aaaa aaaa	60	PTHVB
ERROR	THREATS-BM ASSUME ALTBURST aaaa aaaa	40	PTHVB
9101	DATA-NAME EXPECTED, BUT NOT INPUT. SYNTAS ERR.		PCONT
9102	NAME NOT A VALID DATA NAME, MISSPELLED		PCONT
9110	UNEXPECTED END OF DATA		PCTON
9103	VALUE MUST BE A USER-NAME OR CHAR. STRING		PCONT
9104	TERRAIN VALUE MUST BE A CHAR. STRING		PCONT
9105	TERRAIN VALUE INVALID, SEE SPECIFIC LIST		PCONT
9106	NO VALID DATA-NAMES IN THIS SECTION, NO DSR		PCONT
9101	NO EQUAL SIGN FOLLOWING DATA NAME LOCATION		LOC
9102	NO LEFT PAREN FOLLOWING DATA NAME LOCATION		LOC
9103	ITEM FOLLOWING FIRST LEFT PAREN IS ILLEGAL		LOC
9104	UNBALANCED PARENTHESIS		LOC
9105	ITEM FOLLOWING LAT IS INVALID		LOC
9106	UNBALANCED PARENTHESIS AFTER LATITUDE		LOC
9107	UNBALANCED PARENTHESIS AFTER ALTITUDE		LOC
9108	SYNTAX ERROR FOLLOWS LONGITUDE		LOC
9109	UTM HAS ODD NUMBER OF DIGITS. SEE CODE AT 3030		LOC
9110	UTM-BCDB FOUND ERROR IN EASTING NUMERICS		LOC
9111	UTM-BCDB FOUND ERROR IN NORTHING NUMERICS		LOC
9112	NAME FOLLOWING UTM IS NOT MSL CR HAT		LOC
9113	SYNTAX ERROR FOLLOWS UTM		LOC
9114	UNBALANCED PARENTHESIS FOLLOWS UTM		LOC
9115	GEODETIC TO UTM CONVERSION ERROR		LOC
9117	UTM TO GEODETIC CONVERSION ERROR		LOC
9116	LAT/LONG IS OUTSIDE UTM ZONE OF TERRAIN		LOC

SECTION 3
INPUT EDITOR PHASE C

3-1 OVERVIEW

Final processing of the scenario input data is performed by the Phase C program, INEDC, of the Input Editor. The DIRs produced constitute a complete and easily accessible data base for the application programs. This data base reduces the input and output processing which needs to be performed by the application programs. The types and formats of DIRs described in Section 2 are retained, but the DIRs are updated to incorporate data generated by the INEDC program. The processing performed by INEDC on the DIRs can be summarized as:

(1) Generating pointer addresses for data entities,

(2) Converting data references to address pointers, and

(3) Producing summary data to control the usage of the DIRs by application programs.

3-2 INPUTS

Data is transferred to the INEDC program through the input file designated as TAPE7.

3-2.1 TAPE7 - DIR Input File

TAPE7 contains the DIRs produced by Phase B of the Input Editor. This file must be cataloged by the Phase B program if INEDC is not run in the same job. It is normally in the form of an unformatted binary disk file.

3-3 PROCESSING - INEDC PROGRAM

Program INEDC processes the input DIRs in two passes. The first pass through the input DIRs identifies the data entities and counts them. The specific processing to be done on each DIR is determined by the format and type of the DIR. The subroutines P1CELL and P1CHAR perform the first pass processing.

The second pass through the DIRs is performed by the subroutines P2CELL and P2CHAR. The second pass replaces data references by entity names to data references by record pointers. The use of pointer values in

place of the entity names reduces the volume of data which must be transferred from the INEDC program. The resultant DIRs are used by the application programs and Phase P of the Input Editor.

3-4 OUTPUTS

Data is transferred from the INEDC program by three logical files. These files are designated as TAPE2, TAPE6, and TAPE9. TAPE2 and TAPE6 are used to produce printed output of an informative nature, while TAPE9 is used as the DIR output to the application programs.

3-4.1 TAPE2 - Run and Error Messages File

TAPE2 contains the run and error messages produced by the INEDC program. It is printed by repositioning TAPE2 to the beginning and copying the file contents to the system's output file. This is accomplished through job control cards and is normally done for each run of the INEDC program.

3-4.2 TAPE6 - System Output File

TAPE6 is the logical file name for the system's output file. It contains the error messages produced by the operating system and will be an empty file when the INEDC program is successfully executed.

3-4.3 TAPE9 - DIR Output File

TAPE9 contains the completely processed DIRs from the INEDC program and is the data base to be utilized by the application programs of the ETC^3 model. The DIRs are written in unformatted binary form and is usually saved as a disk file.

3-5 PROGRAM DOCUMENTATION - INEDC

3-5.1 Program INEDC

COMMON BLOCKS: ZZIECA

ZZIECB

ZZIECD

ARGUMENTS: TAPE2, TAPE6, TAPE7, TAPE9

TAPE2	Print output file for run and error messages from INEDC
TAPE6	Output print file
TAPE7	Input file containing unformatted binary DIRs
TAPE9	Output file containing unformatted binary DIRs

This the main routine for the Input Editor Phase C program. The program begins by getting the current date and time of day. The DIR input file, TAPE7, is positioned for processing. The first record on TAPE7 is the date and time from the CONTROL section of the scenario input data. The first pass processing of the input DIRs is performed after the date and time of the input data and the INEDC program execution are printed.

The program will process records having format numbers 1040 and 1100 by calling subroutines P1CHAR and P1CELL, respectively. These subroutines are called for processing until a different record format number is encountered or an EOF on TAPE7 is encountered.

After the first pass processing by subroutines P1CHAR and P1CELL on the DIRs, TAPE7 is repositioned to the beginning and the second pass through the data base is initiated. Calls to subroutines P2CHAR and P2CELL complete the second pass processing.

3-5.2 Program Commons - INEDC

COMMON BLOCKS: ZZIECA
ZZIECB
ZZIECD
ZZUNIT

(1) Common ZZIECA contains tables and data for INEDC. Constants in this common are set by the block data program.

VARIABLE	TYPE	CONTENT
MFMT	Integer	Format number for DIR
MTYP	Integer	Type number for DIR
MCNT	Integer	Continuation number for DIR
MLEN	Integer	Number of words in DIR
INPUT	Integer	DIR input file to INEDC
RCODE	Integer	Return code set by PICHAR and PICELL
NWIDL	Integer	Maximum length of IDL tables in common ZZIECB
NWCODE	Integer	Maximum number of sections defined
JC	Integer	Total number of entries in IDL tables
JEC	Integer	Index of the last characteristic in the IDL table
NAST	Integer	Total number of THREATAST entities
NBM	Integer	Total number of THREATBM entities
TCODE	Integer	Data concerning each section in the characteristic sections
NCNET	Integer	Communication network count
NBUF	Integer	DIR buffer
CBUF	Real	DIR buffer equivalenced to NBUF

(2) Common ZZIECB contains the identifier tables for the entities input contained in the scenario data.

VARIABLE	TYPE	CONTENT
RE	Real	Radius of the earth in meters
PIOVR2	Real	PI divided by 2
D2R	Real	Conversion from degrees to radians
DZERO	Real	Zero
R2D	Real	Conversion from radians to degrees
IDL	Real	First 10 characters of the identifier
IDM	Real	Last six characters of the identifier
TYPEL	Integer	Identifier type code

(3) Common ZZIECD is no longer used by the INEDC progarm.

(4) Common ZZUNIT contains a number of constants and conversion factors.

VARIABLE	TYPE	CONTENT
RE	Real	Radius of earth in meters
PIOVR2	Real	Pi divided by two
D2R	Real	Degrees to radians conversion factor
DZERO	Real	Constant zero
R2D	Real	Radians to degrees conversion factor

3-5.3 Block Data

COMMON BLOCKS: ZZIECA

ZZIECB

ZZIECD

ZZUNIT

The BLOCK DATA program unit supplies constants for the INEDC program's common variables. The reader is directed to the source program listing to determine initial values currently used for the common variables.

3-5.4 Subroutine ARHLOC

COMMON BLOCK: ZZUNIT

ARGUMENTS: F1, G1, H1, F2, G2, H2, AZ, RG

F1	Latitude of point 1
G1	Longitude of point 1
H1	Altitude of point 1 above mean sea level
F2	Latitude of point 2
G2	Longitude of point 2
H2	Altitude of point 2 above mean sea level
AZ	Azimuth of point 1 with respect to point 2
RG	Slant range from point 1 to point 2

ENTRY POINT: LOCARCH

Subroutine ARHLOC computes the latitude and longitude of point 1 given the slant range and azimuth from point 2; and the latitude and longitude of the second point. The altitude of both points are also given.

Entry point LOCARH computes the azimuth and range of point 2 with respect to point 1 from the altitude, latitude, and longitude values of both points.

3-5.5 Subroutine AZRNGE

COMMON BLOCKS: ZZUNIT

ARGUMENTS: F1, F2, RG, AZ, DUMMY

F1	Latitude and longitude of the first point
F2	Latitude and longitude of the second point
RG	Ground range between F1 and F2
AZ	Azimuth of second point with respect to F1
DUMMY	Unused

Subroutine AZRNGE computes the ground range in meters between the points specified by F1 and F2. The coordinates of the points are given in radians. The expression used to calculate the ground range is:

$$RG = 2R_e \sin^{-1} (RG/2R_e)$$

where R_e is the radius of the earth in meters. Both points are considered to be on the earth's surface at mean sea level.

3-5.6 Subroutine BMTVP

COMMON BLOCK: ZZUNIT

ARGUMENTS: PLA, PIA, TIME, V, ANGL, AMJR, BMNR, ECC

- PLA — PLA is the launch point
- PIA — Impact point location
- TIME — Flight time from PLA to PIA
- V — Velocity of the cell in meters per second
- ANGL — Launch angle
- AMJR — Semi-major axis length
- BMNR — Semi-minor axis length
- ECC — Eccentricity of the elipse

Subroutine BMTYP calculates the time of flight for a ballistic missile for a simplified missile path. Some idealizations are: no atmosphere, nonrotating spherical earth, and an average constant velocity. Because the ballistic missile characteristics are used in ETC^3 to lend realism to the scenario and not for trajectory studies, the approximate trajectories are satisfactory.

The equations for ballistic trajectories are derived from the reference "Charts for Determining the Characteristics of Ballistic Trajectories in a Vacuum" by Deane N. Morris, The Rand Corporation, April 1964.

3-5.7 Subroutine CNRAND

ARGUMENTS: CEP, AZIM, RADIUS, XADD, YADD

CEP Circular probable error

AZIM Azimuth

RADIUS Radius corresponding to AZIM

XADD X increment

YADD Y increment

Subroutine CNRAND computes a random azimuth and distance for a given circular error probable from a circular normal distribution.

The azimuth, AZ, is found from the equation:

$$AZ = 2 \; R$$

where R is a random number drawn from a uniform distribution. The value of R is between zero and one exclusive. The radius is computed by calling RANGEN, which returns a value for the variable RAD from a circular normal distribution. The random azimuth and radius values are returned via the variables AZIM and RADIUS. The values for the XADD and YADD variables are computed according to the following equations:

XADD = RADIUS*cos(AZ)

YADD = RADIUS*sin(AZ)

3-5.8 Subroutine COEF

COMMON BLOCK: ZZUNIT

ARGUMENTS: P1, P2, BETA, ANGLH, BMUEL, BMTIM, TATM, TGRD, CM, MRATIO, A, B, E

P1	Launch point location
P2	Impact point location
BETA	Ballistic coefficient of reentry
ANGLH	Launch angle
BMVEL	Vehicle velocity
BMTIM	Time of flight
TATM	Unused
TGRD	Unused
CM	Unused
MRATIO	Unused
A	Semi-major axis from BMTVP
B	Semi-minor axis from BMTVP
E	Eccentricity from BMTVP

Subroutine COEF is used to call subroutine BMTVP. COEF provides an interface between P2CELL and BMTVP. The variables TATM and TGRD are set to dummy values and control is returned to subroutine P2CELL, which is the calling subroutine.

3-5.9 Subroutine PAGE

ARGUMENTS: KK, NUNIT

KK Number of print lines

NUNIT Output unit number

ENTRY POINTS: PAGEA, PAGEB, PAGEC, PAGED

This subroutine controls the pagination and classification marking of the output listings. The following entry points are defined:

(1) PAGE - Controls pagination and is called just before lines are to be printed.

(2) PAGEA - Moves the classification to this subroutine.

(3) PAGEB - Closes the output for NUNIT, and write classification at the bottom of the last page.

(4) PAGEC - Sets up normal data card listing.

(5) PAGED - Sets up messages on unit specified.

3-5.10 Subroutine PICELL

COMMON BLOCKS: ZZIECA

ZZIECB

Subroutine PICELL does the first pass processing of threats for the scenario data. The threat cells are counted and the count is saved in NBM. Cell identifications are not saved because they are not referenced in this processing segment. The threat records are from the THREATBM section of the scenario data.

3-5.11 Subroutine P1CHAR

COMMON BLOCKS: ZZIECA

ZZIEBC

Subroutine P1CHAR performs the first pass processing of the characteristic records. The identification for each characteristic is saved and counted. The identification is the user supplied name which follows the IDENT data name. The subroutine will cause characteristics referenced by the user to be located by transferring control to subroutine SVB. There are specified limits to the number of characteristics which can be processed. An error message is produced if the limit is exceeded and excess characteristics will be ignored.

3-5.12 Subroutine P2CELL

COMMON BLOCKS: ZZIECA

ZZIECB

ZZIECD

ZZUNIT

Subroutine P2CELL performs the second pass processing of the threat cells. Included in the processing done by this subroutine are the resolution of references to other input sections of the scenario input data and calculation of ballistic missile flight times.

3-5.13 Subroutine P2CHAR

COMMON BLOCKS: ZZIECA

ZZIECB

ZZIECD

Subroutine P2CHAR performs the second pass through the characteristics records. These records are identified by having the variable MFMT assigned a value of 1040. The identifications saved from the first pass are used to find the pointers to each reference in a characteristic record. The number of records in each section of the scenario input data is summarized by the count record which precedes the section. The count record contains control information which is used by P2CHAR to establish processing limits.

3-5.14 Function RANDOM

ARGUMENT: X

X Dummy variable

ENTRY POINTS: SEEDIN, OLDRAN

RANDOM is a random number generator which returns values uniformly distributed between the values of zero and one exclusive. Two entry points SEEDIN and OLDRAN control the value used for the seed of the random number generator.

(1) SEEDIN - Sets the seed of the random number generator to X.

(2) OLDRAN - Obtains the current seed of the random number generator and returns the value of the seed in X.

3-5.15 Subroutine RANGEN

ARGUMENTS: XVALUE, YVALUE, KDISTR

XVALUE Standard deviation

YVALUE Random number

KDISTR Distribution to be used

RANGEN generates random numbers from various distributions, however, only the normal distribution is used by ETC[3]. The random number is generated according to the following equations:

$$PROB = R$$

$$YY = XVALUE/LOG_e(2)$$

$$PP = LOG_e(1 - PROB)$$

$$YVALUE = YY * PP$$

R is a uniform random value in the range of 0. to 1. exclusive.

3-6 REFERENCE TABLES-INEDC

Three tables are presented to indicate the way in which the INEDC subprogram units are interrelated with respect to each other and to the common variables. A fourth table is included to identify the entry points of each subprogram unit having multiple entry points. Another table identifies the error messages produced by the INEDC subprograms.

3-6.1 Subroutine Entry Points Table

Table 12 identifies the alternate entry points for each subprogram having multiple entry points.

3-6.2 Subroutine Calls Table

Table 13 identifies the subprogram units called, if any, by each subprogram unit of the INEDC program

3-6.3 Call Structure Table

Table 14 identifies the hierarchical sequence of subprogram calls made by the INEDC program. The call sequences do not reflect any chronological order. Subprogram names appearing in parentheses are entry points into subprograms having multiple entry points.

3-6.4 Common Update Table

Table 15 identifies the variables in the different common blocks which are used or updated by the subprogram units of INEDC. A variable is used if its value is referred to by any executable statement, and is updated if the variable is assigned a value by any DATA or executable statement.

3-6.5 INEDC Error Messages Table

Table 16 lists the error messages which are produced by the INEDC subprogram units. These error messages are produced as an aid to the user and programmer in running the INEDC program. The table identifies the subprogram units which will produce the message.

Table 12. Subroutine entry points.

Subroutine	Entry Points
PAGE	PAGEA
	PAGEB
	PAGEC
	PAGED
ARHLOC	LOCARH

Table 13. INEDC routine calls.

CALLED \ CALLING	ARHLOC	AZRNGE	BMTVP	CNRAND	COEF	INEDC	MVC	PAGE	P1CELL	P1CHAR	P2CELL	P2CHAR	RANDOM	RANGEN	SVB
ARHLOC		X									X				
AZRNGE			X								X				
BMTVP					X										
CNRAND											X				
COEF															
INEDC															
MVC								X							
PAGE						X			X	X	X	X			X
P1CELL						X									
P1CHAR						X									
P2CELL						X									
P2CHAR						X									
RANDOM				X										X	
RANGEN				X											
SVB										X					

Table 14. Call structure.

	0	1	2	3	4	5	6	7
1	INEDC	PAGE	MVC					
2		(PAGEB)	MVC					
3		P1CELL	PAGE	MVC				
4		P1CHAR	PAGE	MVC				
5			SVB	PAGE	MVC			
6		P2CELL	ARHLOC					
7			AZRNGE	(LOCARH)				
8			CNRAND	RANDOM				
9				RANGEN	RANDOM			
10			COEF	BMTVP	AZRNGE	(LOCARH)		
11			PAGE	MVC				
12		P2CHAR	PAGE	MVC				

Table 15. Common update table.

COMMON VARIABLE \ UPDATING ROUTINE	ARHLOC	AZRNGE	BMTVP	CNRAND	COEF	INEDC	MVC	PAGE	P1 CELL	P1 CHAR	P2 CHAR	P2 CHAR	RANDOM	RANGEN	SVB
MFMT						●			●	●	●	●			
MTYP						●			●	●	●	●			
MCNT						●			●	●	●	●			
MLEN						●			●	●	●	●			
INPUT						●									
RCODE									●	●					
NWIDL															
NWCODE															
JC										●					●
JEC										●					
NAST											●				
NBM									●						
TCODE										●					●
NCNET															●
NBUF	●					●			●	●	●	●			
IDL										●					●
IDM										●					●
TYPEL										●					●
CEP												●			
CEPRAD						●									
CEPCNT											●				
BETA												●			

TABLE 16. INEDC error messages.

MESSAGE	ROUTINE
NULL INPUT FILE ON TAPE 7	INEDC
ERROR IEC IN PROCESSING CELLS	INEDC
ERROR IN PROCESSING SYSTEM CHAR	INEDC
SHORT INPUT ON TAPE 7	INEDC
ERROR TOO MANY CHARACTERISTIC RECORDS OF TYPE nn EXCESS IGNORED	P1CHAR
ERROR PASS z CELLS CHARACTERISTICS FOR CELL idt CANNOT BE FOUND. ID THREAT CHAR = idtc	P2CELL
ERROR BM idct HAS INVALID IMPACT TIME	P2CELL
ERROR NAME ida DOES NOT EXIST IN CHARACTERISTICS OF TYPE a DURING PROCESSING OF TYPE NO. 6 ID idb	P2CHAR
ERROR P2CHAR 3190 COUNT ERROR, N=a b ERROR IN SYSTEM CHAR -- PASS 2 COUNT ERROR	P2CHAR
ERROR INEDC SVB 2230 INVALID CHAR NO. nnn	SVB

SECTION 4

INPUT EDITOR PHASE P

4-1 OVERVIEW

The program for Phase P of the Input Editor, INEDP, uses the DIRs from either the Phase B or Phase C programs to produce a formatted listing of data contained in the DIRs. The INEDP program does not perform any processing which would affect DIR usage by application programs. The primary reason for using the INEDP program is to enable manual verification of the DIR file content.

4-2 INPUTS

Data is transferred to the INEDP program through the two logical files designated TAPE9 and TAPE11. TAPE9 contains the DIRs which are produced by either the IEDITB or INEDC programs, and TAPE11 is the Input Editor specification file which is used by the IEDITB program to create DIRs.

4-2.1 TAPE9 - DIR Input File

TAPE9 contains the DIRs produced by either the Phase B or Phase C programs. DIRs are processed according to the format type and not by the generating source. DIRs produced by IEDITB or INEDC are processed in the same way by the INEDP program. The DIRs on TAPE9 are in unformatted binary form, and is normally a cataloged disk file.

4-2.2 TAPE11 - Input Editor Specifications File

TAPE11 contains the data and DIR description specifications necessary to control the conversion of scenario input data into DIRs. The specifications are presently maintained through the usage of the system's utility UPDATE. A more complete presentation of the specifications file is presented in Section 2 and Appendix A.

4-3 PROCESSING

The INEDP program uses the specification file to process the contents of the DIRs in order to present the information in a form suitable for inspection. The processing is controlled by the format of the DIR,

with each format being processed by a separate subprogram unit. The specification file is essential because it permits the transformation of the binary DIRs into the original user input data form. The references to data entities are converted from pointer or coded values back into user recognizable names. The INEDP program performs the inverse operation of the IEDITB and INEDC programs.

4-4 OUTPUTS

Data is transferred from the INEDP program through the logical files designated as TAPE2 and TAPE6. TAPE2 is used to transfer the formatted DIR output, while TAPE6 is used as the system's output file.

4-4.1 TAPE2 - Formatted DIR File

TAPE2 contains the primary output product of the INEDP program. It is printed by repositioning the file to the beginning and copying the TAPE2 contents to the system's output file. This enables printed program outputs to be separated from printed system's output.

4-4.2 TAPE6 - Run and Error Messages File

TAPE6 enables run and error messages detected by the INEDP program or the operating system to be printed. The error messages produced by the INEDP program are identified in 4-6.

4-5 PROGRAM DOCUMENTATION - INEDP

4-5.1 Program INEDP

COMMON BLOCK: ZZDIR

ZZIEP

ZZPAGE

ARGUMENTS: TAPE2, TAPE6, TAPE9, TAPE11

TAPE2	-	Formatted DIR print output file
TAPE6	-	System's print output file
TAPE9	-	Unformatted binary DIR input file
TAPE11	-	Specification data input file

INEDP is the main program for the last phase of the Input Editor. It reads the IEDITB or INEDC produced unformatted binary DIRs and prints a formatted listing for the user. Subroutine RDZD reads the specification file and builds the table to transform DIRs into printable form. Each type of DIR record, which is identified by a format type is processed by a subroutine designed to handle the DIR format.

4-5.2 Program Commons - INEDP

Common Blocks: ZZDIR
ZZIEP
ZZPAGE

There are three common blocks in the INEDP program. These common blocks are described by identifying the variables, the variable type, and a short description of each variable within the common block.

(1) Common ZZDIR contains the variables which are used to describe the DIR.

VARIABLE	TYPE	CONTENT
MFMT	INTEGER	Format number of the DIR
MTYP	INTEGER	Type number of the DIR
MCNT	INTEGER	Continuation number
MLEN	INTEGER	Number of words of NBUF to transfer
NBUF	INTEGER	DIR buffer

(2) Common ZZIEP contains file and error control variables for the INEDP program.

VARIABLE	TYPE	CONTENT
TSP	INTEGER	Specification file designator
TPT	INTEGER	Printed output file designator
TIN	INTEGER	Input DIR file designator
TMS	INTEGER	Error message file designator
NREC	INTEGER	Input DIR record counter
NERR	INTEGER	Error message counter
NSEQ	INTEGER	Entry sequence number within section
NWZ	INTEGER	Index to NF table entry for this data element
NOF	INTEGER	Total number of entries in the current section
NBC	INTEGER	Number of characters in NB
KBIAS	INTEGER	Index to beginning of block in DIR

VARIABLE	TYPE	CONTENT
NENFT	INTEGER	Number of entries in NF table for this DIR format and type
NENDT	INTEGER	Number of ND table entries
IDENT	INTEGER	Record identifier
DATA	INTEGER	IEDITB run date and time
CLASS	INTEGER	IEDITB classification
TITLE	INTEGER	IEDITB title
SECTION	INTEGER	The section being processed
NB	INTEGER	Array containing next item(s) for the print line
NFTA	INTEGER	"A" field contents of card image
NFTB	INTEGER	"B" field contents of card image, or index to ND table
NFTC	INTEGER	"C" field contents of card image
NFTD	INTEGER	"D" field contents of card image
NFTH	INTEGER	Data name or alphanumeric data from card image
NFTT	INTEGER	Card type number
NDTC	INTEGER	Reference number
NDTD	INTEGER	Number of decimal places
NDTG	INTEGER	Units group number

(3) Common ZZPAGE contains the variables which are used to for controlling the input/output format for the PAGE subroutine.

VARIABLE	TYPE	CONTENT
NNPAGE	INTEGER	Data for each input/output file

4-5.3 Block Data

The INEDP program does not use a BLOCK DATA section to set constant values for variables in the common blocks.

4-5.4 Subroutine ERR

COMMON BLOCKS: ZZIEP
ZZDIR

ARGUMENT: KIND

KIND Error number indicator

Subroutine ERR prints error messages for errors encountered by other subroutines in processing DIRs. This subroutine calls subroutine PAGE for line control. The error message is written on the file specified by the variable. The header and first 10 words of the DIR being processed follow on file TMS. Three error messages result in the termination of the INEDP program execution, while all others result in the incrementation of the error counter. Error messages from this subroutine are identified in 4-6.

4-5.5 Subroutine PAGE

COMMON BLOCK: ZZPAGE

ARGUMENTS: NLC, NUC, NRET

NLC Number of lines to be printed

NUC Logical unit numbers

NRET Number of lines left on page

Subroutine PAGE places the data classification on the top and bottom of each page of output. It is called before each WRITE statement to indicate the file being written on and the number of lines.

4-5.6 Subroutine PDW

COMMON BLOCKS: ZZDIR

ZZIEP

ARGUMENTS: KDW, K1, K2, NRET

KDW	Index to DIR buffer array
K1	DIR or continuation block processing flag
K2	Last word in the DIR to be processed
NRET	Return code

Subroutine PDW processes a word in the DIR. The specification tables are checked for the set which pertain to the word. Subroutine PDRAY is called if the word is part of an array and subroutine PDLOC is called if the word is not part of an array.

4-5.7 Subroutine PDLOC

COMMON BLOCKS: ZZDIR

ZZIEP

ARGUMENTS: KDW, K2, KF

KDW	Index to latitude value in the DIR
K2	Unused
KF	Index for the second dimension of array NFTH

PDLOC prints the geographic location in latitude and longitude values. The latitude and longitude are converted from radians to degrees for printing.

4-5.8 Subroutine PDRAY

COMMON BLOCKS: ZZDIR

ZZIEP

ARGUMENTS: KDW, K2, KF

KDW	Index to the array of data in the DIR
K2	Unused
KF	Index to the specification table NF

Subroutine PDRAY prints DIR data stored in arrays. The subroutine can be used to print arrays having up to three dimensions. For three dimensional arrays, the third dimension is printed as the header to a table. One and two dimensional arrays are printed with a single header. The header corresponds to the name of the array.

4-5.9 Subroutine PDRAYQ

COMMON BLOCKS: ZZDIR

ZZIEP

ARGUMENTS: IA, IB, KZX, KZR, KF1, KF2, KM

IA	Dimension number
IB	Dimension value
KZX	Index to the correct ZX entry in table NF
KZR	Index to the specification following the ZX entry to be used
KF1	Initial search index for table NF
KF2	Final search index for table NF
KM	Flag

Subroutine PDRAYQ locates the specification for an array datum. The subroutine searches the NF table from KF1 and KF2 to find the ZX specification for the datum in table NF.

4-5.10 Subroutine PDDATA

COMMON BLOCKS: ZZDIR

ZZIEP

ARGUMENTS: KZR, POINT, CHARS, INDEX

KZR	Index to the NF table entry describing the datum
POINT	Value in the DIR
CHARS	Character string for the result of the conversion
INDEX	Index to the last word in CHARS that holds data

This subroutine converts binary data to character data and stores it as part of a character string. The data conversion depends upon the format of the datum. Data having units of radians or meters are converted to degrees and kilometers, respectively.

4-5.11 Subroutine PUT

COMMON BLOCK: ZZIEP

ARGUMENT: MODE

MODE Processing mode

Subroutine PUT places information in the output line. The information controlling the placement of information in the output line is transmitted to PUT by the MODE variable. There are five values for MODE, each of which results in a special output processing function.

(1) MODE = 1 causes the current line to be printed and a new line started with the current input data.

(2) MODE = 2 causes data names to be aligned in three columns in the print line.

(3) MODE = 3 causes a blank line to be printed, it does not change the contents of the current line.

(4) MODE = 4 causes a new page to be started on TAPE2 which is the program output file.

(5) MODE = 5 causes the current line to be printed and a new line started.

4-5.12 Subroutine QAIRSP

COMMON BLOCKS: ZZDIR

ZZIEP

ARGUMENT: NRET

NRET Return code from QAIRSP

Subroutine QAIRSP prints the airspace description records of the scenario data. Airspaces are used by the wargame simulation models to describe various types of SAM weapon control zones. The titles and column headings are printed on a new page. The type of aircraft defined for the airspace are printed from the array LIST. This array is keyed to the type field in the threat characteristic DIR.

4-5.13 Subroutine Q1024

COMMON BLOCKS: ZZDIR

ZZIEP

ZZPAGE

ARGUMENT: NRET

NRET Return code from Q1024

Subroutine Q1024 prints the information contained in the CONTROL section of the scenario data. The control information printed consists of the IEDITB execution date and time, the title given to the data base, the security classification of the data base, and the terrain data.

4-5.14 Subroutine Q1040

COMMON BLOCKS: ZZDIR

ZZIEP

ARGUMENT: NRET

NRET Return code for subroutine Q1040

Subroutine Q1040 reads and processes all DIRs which have MFMT value of 1040 or characteristics DIRs. The subroutine will initiate the processing necessary to convert the binary DIRs into a printable character string and to place the character string into an output line by calling subroutines PDW and PUT, respectively. The DIRs, blocks and continuation record are all processed by the subroutine.

4-5.15 *Subroutine Q1080*

COMMON BLOCKS: ZZDIR

ZZIEP

ARGUMENT: NRET

NRET Return code

Subroutine Q1080 prints information concerning the communication lines.

4-5.16 Subroutine Q1100

COMMON BLOCKS: ZZDIR

ZZIEP

ARGUMENT: NRET

NRET Return code

Subroutine Q1100 prints all data concerning the track of ballistic missiles in the scenario data. The missile characteristics and the launch and impact locations are printed by the subroutine. The locations are converted from radians to degrees of latitude and longitude before being printed. A title is printed to identify the output.

4-5.17 Subroutine RDZD

COMMON BLOCKS: ZZDIR

ZZIEP

Subroutine RDZD reads from the specification file all the ZD card images. The data from these cards are saved in the ND tables. The specification records are read from file TSP.

4-5.18 Subroutine RDZF

COMMON BLOCKS: ZZDIR

ZZIEP

Subroutine RDZF reads the specification file and finds the set of specifications which describe the DIR type. The set is stored in table NF. Currently, only DIRs having an MFMT value of 1040 is processed using RDZF.

4-6 REFERENCE TABLES - INEDP

Three tables are presented to indicate to the programmer the way in which the INEDP subprogram units are interrelated with respect to each other and to the common variables. A fourth table identifies the error messages produced by the INEDP program and the subprogram unit responsible for detecting the error condition.

4-6.1 Subroutine Call Table

Table 17 identifies the subprogram units called, if any, by each subprogram unit of the INEDP program.

4-6.2 Call Structure Table

Table 18 identifies the hierarchical sequence of subprogram calls made by the INEDP program.

4-6.3 Common Update Table

Table 19 identifies the variables in the various common blocks which are used or updated by the INEDP subprogram units. A variable in common is used by a subprogram unit if the common variable is referenced by any executable statement, and is updated if is assigned a value by any DATA or executable statement.

4-6.4 Error Messages

Table 20 lists the error messages which are produced by the INEDP subprogram units. These error messages are printed to aid the user and programmer in running the INEDP program. The subroutines for detecting the error conditions are provided in the table also.

Table 17. Subroutine call.

CALLED \ CALLING	EQC	ERR	INEDP	MVC	PAGE	PDDATA	PDLOC	PDRAY	PDRAYQ	PDW	PUT	QAIRSP	Q1024	Q1040	Q1060	Q1080	Q1100	RDZD	RDZF
EQC			X									X							
ERR			X																
INEDP																			
MVC						X													
PAGE										X									
PDDATA								X											
PDLOC										X									
PDRAY										X									
PDW														X					
PUT							X	X											
QAIRSP															X				
Q1024			X																
Q1040			X																
Q1060			X																
Q1080			X																
Q1100			X																
RDZD			X																
RDZF														X					

Table 18. Call structure.

	0	1	2	3	4	5	6	7
1	INEDP	ERR	PAGE					
2		PAGE						
3		Q1024	ERR	PAGE				
4			PAGE					
5		Q1040	ERR	PAGE				
6			PAGE					
7			POW	PAGE				
8				PDLOC	PUT	MVC		
9						PAGE		
10				PDRAY	PAGE			
11					PDDATA	MVC		
12					PDRAYQ			
13					PUT			
14						PAGE		
15			PUT	MVC				
16				PAGE				
17			RDZF	ERR	PAGE			
18		Q1060	ERR	PAGE				
19			PAGE					
20			QAIRSP	EQC				
21				MVC				
22				PAGE				
23		Q1080	ERR	PAGE				
24			PAGE					
25		Q1100	ERR	PAGE				
26			MVC					
27			PAGE					
28		RDZD	ERR	PAGE				

Table 19. Common update table.

COMMON BLOCK	VARIABLE \ ROUTINE	EQC	ERR	INEDP	MVC	PAGE	PDDATA	PDLOC	PDRAY	PDRAYQ	PDW	PUT	QAIRSP	Q1024	Q1040	Q1060	Q1080	Q1100	RDZD	RDZF
	MFMT	-	0	x	-	-	-	-	-	-	0	-	x	-	x	x	x	x	-	-
	MTYP		0	x							0		x	0	x	x	x	x		
ZZDIR	MCNT		0	x									x		x	x	x	x		
	MLEN		0	x									x		x	x	x	x		
	NBUF		0	x			0	0	0		0		x	0	x	x	x	x	x	
	TSP			x																0
	TPT			x				0	0			0	0			0	0	0		
	TIN			x									0		0	0	0	0		
	TMS			x							0									
	NREC			x											x	x		x		
	NERR		x	x																
	NSEQ								0			0			x					
	NWZ										x									
	NOF								0			0			x					
	NBC										x	0	x							
	KBIAS										0				x					
	NENFT								0		0									x
	NENDT																		x	0
	IDENT								0			0			x					
ZZIEP	DATE													x						
	CLASS					x														
	TITLE													x						
	SECTION								0			0								x
	NB										x	x								
	NFTA								0	0	0									x
	NFTB						0		0	0	0									x
	NFTC								0		0									x
	NFTD								0											x
	NFTH							0	0			0								x
	NFTT						0		0	0	0									x
	NDTC																		x	0
	NDTD						0				0								x	
	NDTG						0				0								x	
ZZPAGE	NNPAGE			x		x								x						

x = UPDATES

0 = REFERENCES

Table 20. Error messages.

ERROR MESSAGES	ROUTINE
END OF INEDP RUN, DIR COUNT = nnnn, NO. ERRORS = nnnn	INEDP
ERROR INEDP aaaa aaaa ERROR NO. nnnn (SEE TABLE)	ERR
DIR IS nnnn nnnn nnnn nnnn oooo oooo oooo oooo	ERR
ERROR IN PDW, INVALID TABLE ENTRY nnnn nnnn	PDW
INPUT RECORD HAS INVALID MFMT.	ERR
UNEXPECTED EOF WHILE READING FILE TSP.	ERR
TABLE NDT HAS OVERFLOWED.	ERR
CARD IMAGE 2xx IS INVALID IN RDZD.	ERR
INPUT RECORD HAS INVALID MTYP.	ERR
CARD IMAGE 2xx IS INVALID IN RDZF.	ERR
TABLE NFT HAS OVERFLOWED.	ERR
NO MATCH IN NDT-NFT TABLES.	ERR
SECTION MISSING IN NFT FILE.	ERR

SECTION 5
DAMAGE

5-1 OVERVIEW

A number of functional and physical nuclear weapons effects for a hypothesized C^3 system are calculated by the DAMAGE program. Each effect is normally computed by a separate subroutine within the program in order to improve the modularity of the application program. Thus, the effects which can be computed are dependent on the generated executable program module. The subroutines described in this section reflect the nuclear weapons effect previously documented in reference 1 of Appendix B.

The DIR's produced by the Phase C Input Editor program are used as the primary source of scenario laydown and C^3 systems data for the DAMAGE program. This application program computes the nuclear effects as determined by analytic expressions from EM-1 and other sources and compares the results to damage threshold values specified by the user. When the condition represented by the threshold value exists for the C^3 system component, the component is indicated as having failed due to the effect. These effect values are computed for the equipment, lines, nodes, and sites of the C^3 system and determine their failure status for the simulation.

5-2 INPUTS

Data is transferred to the DAMAGE program through a single file; it is designated as TAPE20.

5-2.1 TAPE20 - Input DIR File

TAPE20 contains the DIRs produced by Phases B and C of the Input Editor. This file is cataloged by the Phase C program as an unformatted binary file.

5-3 PROCESSING - DAMAGE PROGRAM

Program DAMAGE processes the nuclear weapons effect on the communications' lines and sites in a recursive manner. All data from the DIRs describing the C^3 system are stored by the program. The DAMAGE program computes all the nuclear effects from each burst for each line and site in the C^3 system. There are eleven nuclear effect measures which are computed

by the DAMAGE program. These effects are summarized herein and further in references 1 and 3 of Appendix B. These effects are:

(1) Acoustic reverberation
(2) Dynamic pressure
(3) Gamma radiation flux
(4) Gamma radiation
(5) High-altitude electromagnetic pulse radiation
(6) Low-altitude electromagnetic pulse radiation
(7) Neutron radiation
(8) Overpressure
(9) Thermal radiation
(10) Vulnerability number
(11) X-ray radiation

Computation of these effect measures for each line or site which is designated as hardened against the measure is bypassed, otherwise the appropriate effect *subroutine will be called for the* burst and target pair. The processing continues until all records in the DIR input file have been read and processed.

5-4 OUTPUTS

Data is transferred from the DAMAGE program though two logical files. These files are designated as TAPE6 and TAPE21. TAPE6 is used to transfer DAMAGE execution results and messages, while TAPE21 is used to create an output file in DIR format for a plotting program.

5-4.1 TAPE6 - Output File

TAPE6 is the logical file name for the system's output file. It contains the results calculated by the DAMAGE program as well as messages for error conditions detected by the program during execution. The contents of this file are always printed.

5-4.2 TAPE21 - DIR Output File

TAPE21 contains selected DIRs, in the exact format as the input DIRs, which are used for plotting purposes. The file consists of unformatted binary DIRs and needs to be cataloged if it is to be used in the production of plots. The plots are done by external programs.

5-5 PROGRAM DOCUMENTATION - DAMAGE

5-5.1 Program DAMAGE

COMMON BLOCKS:

ZZDIR

ZZDMG

ZZEVNT

ARGUMENT: TAPE5, TAPE6, TAPE20, TAPE 21

TAPE5 - TAPE5 is equivalenced to the system's input file.

TAPE6 - TAPE6 is equivalenced to the systems' output file. It is used to print the results of the DAMAGE execution and any messages issued by the program to the user.

TAPE20 - TAPE20 is the input file containing the DIRs produced by the successive application of the Phase B and Phase C programs of the Input Editor. *The DIRs are transferred as unformatted* binary records to the DAMAGE program.

TAPE21 - TAPE21 is an output file consisting of selected DIRs. The information from these DIRs is used by another program to produce plots.

The DAMAGE application program calculates a number of nuclear effects on communications lines and sites for each burst contained in the DIR file. The calculation of a particular effect is normally accomplished by calling a single independent subroutine. The current convention for these subroutines is to have all data transfer through the use of argument variables and not common variables. These subroutines are readily incorporated into or removed from the DAMAGE program in order to facilitate the use of the model as a tool in performing systems level assessment of nuclear effects on components of various communication systems.

Nuclear effects which are to be examined are calculated by the program for each nuclear burst-site and burst-communication line pair. The

program accomplishes the nuclear effects computation first by site and then by communication line.

5-5.2 Program Commons-DAMAGE

COMMON BLOCKS: ZZDIR

ZZDMG

ZZEVNT

There are three common blocks in the DAMAGE program. These common blocks are described by identifying the variables, the variable type, and by providing a short description of the variable content.

(1) Common ZZDIR contains the variables which describe the DIRS. These DIRS contain the scenario input data.

VARIABLE	TYPE	CONTENT
FILEMS	Integer	FILE of formatted error messages
FILEIE	Integer	FILE of the input editor (unformatted)
MXSITE	Integer	Number of sites in the problem
MXLINE	Integer	Number of communication lines in the problem
YTC	Real	Yield of threat (YT)
KSTC	Integer	Pointer to site characteristics
FST	Real	Latitude of sites
GST	Real	Longitude of sites
HST	Real	Altitude of sites
ISTID	Integer	ID site characteristics
ESC	Real	Nuclear effect level limit
RNE	Real	Resulting nuclear effects value
JNE	Integer	Have RNE been calculated? (0 or 1)
DEFALT	Real	Nuclear weapons effects default value (see Table 2, Input Data Summary)
INEID	Integer	Nuclear effect identifiers
NML1	Integer	Pointer to node 1 site
NML2	Integer	Pointer to node 2 site
IMLID	Integer	Communication characteristics Identifier
EMY	Real	Nuclear effect level on communication line

VARIABLE	TYPE	CONTENT
KMLC	Integer	Pointer to communication line characteristics
MYTYP	Integer	Communication line type (1-9)
CMYFRQ	Real	Frequency associated with transmitter
CRDB	Real	Decibel (Db) level for receiving site
FTH	Real	Burst Latitude
GTH	Real	Burst Longitude
HTH	Real	Height of burst
YTH	Real	Yield
RB	Real	Fireball radius
THB	Real	Burst size
HB	Real	Burst altitude
NSCID	Integer	Identifier of site characteristics
MYID	Integer	Communication line characteristics identifier
NSPOTS	Real	Sun spot numbers
SUNZEN	Real	Sun zenith angle
LDSITE		Unused
CMLOUT	Real	Com line out
TB	Real	Time of burst (sec)
NUMBST	Integer	Number of bursts (threats)
MFMT	Integer	Format number
MTYP	Integer	Type number
MCNT	Integer	Continuation number
MLEN	Integer	Length (words in NBUF)
NBUF	Integer	Data buffer
TC	Argument	Threat characteristics
ST	Argument	Site characteristics
ML	Argument	Communication line
MY	Argument	Communication line characteristics
TH	Argument	Threats

(2) Common block ZZDIR contains the variables which are actually used to transfer the DIRs into the DAMAGE program.

VARIABLE	TYPE	CONTENT
MFMT	INTEGER	DIR format number
MTYP	INTEGER	DIR type number
MCNT	INTEGER	DIR counter
MLEN	INTEGER	DIR length indicator
NBUF	INTEGER	DIR buffer

(3) Common block ZZEVNT contains the variables which describe the DIR data as it is stored in the DAMAGE program tables

VARIABLE	TYPE	CONTENT
TT	REAL	Event time
TTT	REAL	Temporary event time
NC	INTEGER	Cell sequence number
NCT	INTEGER	Temporary sequence number
ND	INTEGER	Definite sequence number
NDT	INTEGER	Temporary definite sequence number
NF	INTEGER	Definite entity sequence number
NFT	INTEGER	Temporary definite entity sequence number
FT	INTEGER	Entity type
NTT	INTEGER	Temporary entity type
ITY	INTEGER	Event type
ITYT	INTEGER	Temporary event type

(4) Common block BURST contains the bursts of the scenario input data.

VARIABLE	TYPE	CONTENT
ARRAY	REAL	Nuclear weapons blast records

5-5.3 Block Data DMGBLK

The DMGBLK unit initializes the variables which are either single variables or arrays, for the common block ZZDMG. The specific variables which have values set for them are MXSITE, MXLINE, DEFALT, INELD, and the reader is referred to a program listing for the exact data values being assigned.

5-5.4 Subroutine ACT

ARGUMENTS: R, YIELD

R	Slant range from the burst in meters
YIELD	Weapon yield in kilotons

Subroutine ACT predicts the reverberation noise level from a nuclear burst as a function of range and signal frequency. The calculated noise level is compared by the analyst to the overload threshold of a given sonar (the frequency of the noise must be within the sonar's bandwidth). The subroutine does not return any values to the DAMAGE program, but prints the noise levels in decibels (dB) referenced to 1 Pa/Hz for C W signals of frequencies 1000 ,2000, 3500, 5000, and 7500 Hz. The values are printed for times of 50 and 200 seconds after the burst. Also printed are the ranges between the burst and points of interest, and the time of signal arrival at the points of interest. The statement:

BEYOND RANGE OF APPLICABILITY

is printed when the slant range between the burst and points of interest exceed 111 kilometers.

5-5.5 Subroutine ARHLOC

COMMON BLOCK: ZZUNIT

ARGUMENTS: F1, G1, H1, F2, G2, H2, AZ, RG

F1	Latitude of point 1
G1	Longitude of point 1
H1	Altitude of point 1 above mean sea level
F2	Latitude of point 2
G2	Longtitude of point 2
H2	Altitude of point 2 above mean sea level
AZ	Azimith of point 1 with respect to point 2
RG	Slant range from point 1 to point 2

Entry Point: LOCARH

Subroutine ARHLOC computes the latitude and longitude of point 1 given the slant range and azimuth from point 2; and the latitude of the second point. The altitude of both points are also given.

Entry point LOCARH computes the azimuth and range of point 2 with respect to point 1 from the altitude, latitude, and longitude measures of both points.

5-5.6 Subroutine CALPSI

ARGUMENTS: D, Y, H, PRF

D	Ground range in meters
Y	Weapon yield in kilotons
H	Height of burst in meters
PRF	Overpressure in psi

ENTRY POINT: INVPSI

Subroutine CALPSI calculates the overpressure by performing a table look up process using linear and logarithmic interpolation. The data for the table are derived from the July 1972 edition of EM-1, Reference 4 of Appendix B. The calculated pressure is returned to the DAMAGE program by the argument PRF. The dynamic pressure must be computed prior to the computation of overpressure which is done by subroutine DYNPSI.

Entry point INVPSI is used to calculate the ground range from the burst to the point of interest given the dynamic pressure, weapon yield and the height of burst.

5-5.7 Subroutine DCNTL

Common Blocks: Blank

ZZDMG

ZZDIR

ARGUMENT: NCODE

NCODE Return Code

Subroutine DCNTL reads the control records of the DIR input file and sets the argument variable NCODE.

5-5.8 Subroutine DSYS

Common Blocks: Blank
ZZDMG
ZZDIR

ARGUMENT: NCODE

NCODE Return Code

Subroutine DSYS reads the DIRs and stores the data into tables.

5-5.9 Subroutine DYNPSI

ARGUMENTS: ROV, DROV

ROV	Overpressure in psi
DROV	Peak dynamic pressure

Subroutine DYNPSI calculates the dynamic pressure from the overpressure value calculated by the CALPSI subroutine. The equation, used to compute the peak dynamic overpressure is reproduced below:

$$DROV = 2.5 \times ROV^2/(102.9 + ROV)$$

The calculated value is returned to the DAMAGE program through the use of the argument DROV.

5-5.10 Subroutine FSTQ

ARGUMENTS: R, YIELD, HEIGHT, FSTFDW, FSTFDC, BFD, ERFD

R	Slant range from detoration in meters
YIELD	Weapon yeild in kilotons
HEIGHT	Site or equipment elevation in meters
FSTFDW	Calculated electromagnetic field valu over water
FSTFDL	Calculated electromagnetic field value over land
BFD	Calculated electromagnetic field value
ERFD	Calculated electromagnetic field value

Subroutine FSTQ calculates low altitude electromagnetic pulse effects on sites and equipment. The expressions used were derived by HDL in Reference 5 of Appendix B. The expression generates free field low altitude EMP values for surface bursts.

5-5.11 Subroutine HAEMP

ARGUMENTS: LATB, BLONGB, LATO, OLONGO, HOBO, EFT

LATB	Latitude of burst location
BLONGB	Longitude of burst location
LATO	Latitude of observer location
OLONGO	Longitude of observer location
HOBO	Height of burst
EFT	Electric field ratio at the observer location

Subroutine HAEMP calculates the ratio of the electric field at the observer location to the maximum electric field.

5-5.12 Subroutine NEUTR

ARGUMENTS: R, YIELD, HOB, XNEUTR

R	Slant range in meters
YIELD	Weapon yield in kilotons
HOB	This argument is not currently being used
XNEUTR	Neutron radiation in N/cm^2

ENTRY POINT: INVNEUT

Subroutine NEUTR calculates the neutron radiation density. The computed value is returned to the DAMAGE program through the argument XNEUTR. The two statements below are printed as informative messages to warn the DAMAGE program user of invalid results.

(1) OUTSIDE LIMITS OF APPLICABILITY, SLANT RANGE .LT. 60W(1/3), RANGE = NNN METERS, YIELD = NNNN. KT

(2) OUTSIDE LIMITS OF APPLICABILITY, FLUENCE TOO LARGE, XNEUTR = NNNN. N/CM2

5-5.13 Subroutine TDOSE

ARGUMENTS: R, YIELD, HOB, TD

R	Slant range in meters
YIELD	Weapon yield in kilotons
HOB	Argument is currently not used
TD	Calculated total dose in Rads

Subroutine TDOSE calculates the total gamma dosage and returns the calculated dosage through the argument TD. It also prints two messages to warn the DAMAGE program user that the resultant value is invalid. The two messages printed are:

(1) OUTSIDE LIMITS OF APPLICABILITY, SLANT RANGE .LT. 60 W(1/3), RANGE = NNNN. METERS, YIELD = NN. KT.

(2) OUTSIDE LIMITS OF APPLICABILITY, TOTAL DOSE TOO LARGE, TDOSE = NNNN. RADS.

The appearance of only the first message will not enable the user or the programmer to trace down the print initiating subroutine unless other information is available. The first message is produced by several subroutines.

5-5.14 Subroutine THRMAL

ARGUMENTS: HOB, R, YIELD, Q

HOB	Height of burst in meters
R	Slant range in meters
YIELD	Weapon yield in kilotons
Q	Radiant exposure in cal/cm^2

ENTRY POINT: INVTRML

Subroutine THRMAL calculates the radiant exposure according to equations given in Chapter 3 of EMA, reference 4 of Appendix B. The subroutine returns the calculated radiant exposure value through the argument Q. The equations used to calculate the thermal partition function and transmittance are also from EM-1.

Entry point INVTRML is used to calculate the slant range given the radiant exposure and the weapon yield.

5-5.15 Subroutine TRSODG

ARGUMENTS: R, YIELD, H, TRSDG

YIELD	Nuclear weapon yield in kilotons
H	Height of burst
TRSDG	Calculated dosage in rads/sec

ENTRY POINTS: INVGD

Subroutine TRSODG calculates a value for the argument TRSDG, gamma rate, according to equations from reference 4. The subroutine can print three messages to warn the DAMAGE program user of the non-applicability of the subroutine to perform computations on the input data.

(1) OUTSIDE LIMITS OF APPLICABILITY, RANGE TOO LARGE, RANGE = NNN. METER

(2) OUTSIDE LIMITS OF APPLICABILITY, SLANT RANGE, LT. 60W(1/3), RANGE = NNN. METERS, YIELD = NNN. KT)

(3) OUTSIDE LIMITS OF APPLICABILITY, DOSE RATE TOO LARGE, TRSDG = NNN. RADS/sec.

Entry point INVGD is used to calculate the slant range given the weapon yield and dosage rate assigned the argument variable TRSDG.

5-5.16 Subroutine XRAY

ARGUMENTS: LATB, LONGB, HBO, LATS, LONGS, HSO, W, FI

LATB	Latitude of the burst
LONGB	Longitude of the burst
HBO	Height of burst in meters
LATS	Latitude of the point of interest
LONGS	Longitude of the point of interest
HSO	Height of point of interest in meters
W	Weapon yield in kilotons
FI	X-ray fluence level in cal/cm^2

Subroutine XRAY calculates the x-ray radiation according to equations from EM-1, reference 4. The subroutine returns the calculated value to the DAMAGE program through the argument FI.

5-5.17 Subroutine VULN

ARGUMENTS: NN, KN, YIELD, RES

NN	First numeric portion of vulnerability number
KN	Second numeric portion of vulnerability number
YIELD	Nuclear weapon yield
RES	Overpressure or dynamic pressure

Subroutine VULN calculates the dynamic pressure or overpressure for a vulnerability number expressed as NN-P-KN or NN-Q-KN. If the vulnerability number is of the last form the overpressure is calculated, otherwise the dynamic pressure is calculated. The NN-Q-KN form of the vulnerability number is indicated by making the argument NN negative.

5-6 REFERENCE TABLES-DAMAGE

Due to the modular structure of the DAMAGE program and subprogram units only one table is presented for this section. The single table used to summarize information in this section is the subroutine entry point table.

5-6.1 Subroutine Entry Point Table

Table 22 shows the alternate entry pints to the various subroutines of the DAMAGE program. In all cases, the entry point cause the inverse of the processing to be done when the subroutine is entered by the subroutine name.

5-6.2 Subroutine Calls

The modular structure of the DAMAGE program eliminates subroutine to subroutine calls except to FORTRAN library subroutine. The DAMAGE program performs all the calls to the subroutines documented. Because of the simple structure, a table is not included.

5-6.3 *Call Structure*

A call structure table is not presented for the same reason as in paragraph 5-6.2.

5-6.4 Common Update

The update of variable in the common blocks are performed by the DAMAGE program and the two subroutines DCNTL and DSYS, which are used to process the input DIRs. These three program units perform all the accessing of common block variables, since communication with all other subroutines are through call arguments.

5-6.5 *Error Messages*

The DAMAGE program produces the two error messages which are reproduced below. The subroutines in the program the calculations and return the results to the main program. The two messages produced are listed below and both are from subroutine DSYS.

DMG003	ERROR IN DSYS
DMG003	ERROR IN DSYS NO THREATS IN IE FILE

Table 21. Subroutine entry points.

SUBROUTINE	ENTRY POINT
ARHLOC	LOCARH
CALPSI	INVPSI
NEUTR	INVNEUT
THRMAL	INVTRML
TRSODG	INVGD

Appendix A
Specification Card Formats

ZX Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZX	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Index Name
4	31-40		
5	41-50		Index Value

When the index name is encountered, the index value for that dimension is set until it is overridden by a previously set dimension. If the ZX card is followed by a ZI or ZR card, that card specifies the ZD card to process the Data.

If the index value is negative, the corresponding index is set to one(1) initially, and incremented by the absolute value of the index value each time it is encountered until a previously named index is used. At this point, it will be reintialized. If there are more than one ZX's with a given index name, all the ZXs are processed in the order in which they appear in the array specification.

Warning: A dummy ZX must appear after each ZX with duplicate index names if ZFs are used with that index name on subsequent cards in the specification.

PRECEDING PAGE BLANK-NOT FILMED

ZR Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZR	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Data-Name
4	31-40		Location
5	41-50		ZD Card Reference Number
6	51-60		Maximum List Length (Default is 1)
7	61-70		Minimum List Length (Default is 0)

The data-name acts as a trigger. When it is encountered, numeric entries are read, subjected to the processing indicated by the referenced ZD entry and stored in the DIR until the maximum list length value is exceeded or a non-numeric entry is encountered. The ith entry is stored in location i-1. If the maximum list length is less than 0, the ith entry is processed and stored according to the (i-1)st card following the ZI card. Data is stored as a real number.

ZT Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZT	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		ZFD Type Section To Be Processed Next.

ZA Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZA	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Data-Name
4	31-40		Location for Characters (Default is no storage)
5	41-50		Number of Characters To Be Moved And/Or Compared
6	51-60		Location of Value Index (Default is No Index)
7	61-70		Special Processing If Not Zero (Default is No Processing)

If field 4 is not zero, the data-name following the trigger will be moved to the location specified.

If field 5 is non zero, but field 6 is zero, the data-name is searched for among the ZL cards following, and if it is found, the index values is placed in the location specified by field 6.

If field 5 and field 6 are non zero, a list of data-names is used. Each data-name is searched for among the ZL cards following to obtain the index value. The first data-name may be stored in the location specified.

ZB Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZB	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Blockname
4	31-40		Length
5	41-50		Continuation Flag

If there is a ZI, ZR, or ZA card immediately following the ZB card, or there is a ZI, ZR, ZA, or ZM card with the same name as the block, the corresponding processing is done immediately after the block is utilized. If the continuation flag is 0, the block is put on the current continuation record if it fits; otherwise it is forced to the next continuation record, and it will be the only block that is on the record. If the continuation flag is less than zero, the next *name flag* will be set to zero.

ZBE Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZBE	Z-Card Type Designator

ZD Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZD	Z-Card Type Designator
2	5-10		Reference Number
3	11-20		Units Reference Number
4	21-30		Scale Factor
5	31-40		Minimum
6	41-50		Maximum
7	51-60		Quantum

ZF Card Format

Type 1

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZF	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Section Name
4	31-40		Format
5	41-50		Type
6	51-60		Length (Minimum Length If Blocks are used)
7	61-70		Pointer[1]

[1] 1000 (Pointer To First Predefined Block) + (Maximum Length/Record)

Type 2

Field Number	Card Columns	Field Entry	Field Descriptions
1	1-4	ZF	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Subsection Name
4	31-40	-1.	
5	41-50	-1.	
6	51-60	-1.	
7	61-70	-1.	

ZFD Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZFD	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Subsection Name
4	31-40		

The ZID card in the Define Section must have the same data-name as specified by this card.

ZI Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZI	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Data-Name
4	31-40		Location
5	41-50		ZD Card Reference Number
6	51-60		Maximum List Legth (Default is 1)
7	61-70		Minimum List Length (Default is 0)

The data-name acts as a trigger. When it is encountered, numeric entries are read, subjected to the processing indicated by the referenced ZD entry, and stored in the DIR until the maximum list length value is exceeded or a non-numeric entry is encountered. The i^{th} entry is stored in location i-1. If the maximum list length is less than 0, the i^{th} entry is processed and stored according to the (i-1)st card following the ZI card. Data is stored as an integer number.

ZIA Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZIA	Z-card Type Designator
2	11-26		Initialization Value
3	31-40		First Location to Be Initialized
4	41-50		Number of Characters
5	51-60		Last Location to Be Initialized
6	61-70		Subscript increment

Initializes specified location(s) to Alphanumeric Form of Value

ZID Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZID	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Ident Name
4	31-40		

This item is processed identically with ZA, but it also triggers a new DIR to be written. At least one must be included in each define section.

ZII Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZII	Z-Card Type Designator
2	31-40		First Location to be Initialized
3	41-50		Initialization Value
4	51-60		Last Location to be Initialized (Default is First Location)
5	61-70		Subscript Increment (Default is 1)

1. Initializes Specified Location(s) to integer form of value

ZIR Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	IR	Z-Card Type Designator
2	31-40		First Location To Be Initialized
3	41-50		Initialization Value
4	51-60		Last Location To Be Initialized (Default is First Location)
5	61-70		Subscript Increment (Default is 1)

The ZIR card initializes specified location(s) to real form of value

AD-A077 617 BDM CORP MCLEAN VA F/G 17/2
EUROPEAN THEATER COMMAND, CONTROL AND COMMUNICATIONS (ETC3) MOD--ETC(U)
NOV 78 P BOGUE , R LABASH , J SANDERS DNA001-78-C-0070
UNCLASSIFIED BDM/W-78-718-TR DNA-4351T-2B NL

3 OF 4
ADA
077617

END
DATE
FILMED
1-80
DDC

CONT

ZL Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZL	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Data-Name
4	31-40		Index Value (Type 2)
5	41-50		Index Value (Type 2)
6	51-60		Location Bias (Type 3)
7	61-70		Operation (Type 3)

The index value of field 4 is stored in the location specified by field 6 on the ZA card for fields qualified by type 2 notation.

The index value of field 5 is operated into the location specified by field 6 minus 1 plus field 6 from the ZA card. The operations are:

0 set
1 add (4)
2 or (v)

ZM Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZM	Z-Card Type Designator
2	5-10		Subroutine Reference Number
3	11-26		Data-Name
4	31-40		Array Starting Location
5	41-50		Slowest Incremented Dimension Size
6	51-60		Medium Incremented Dimension Size
7	61-70		Fastest Incremented Dimension Size

If the array is one dimensional fields 6 and 7 should be omitted, and if the array is two dimensional field 7 should be omitted. If any of the three is negative, that index value is determined by the number of times that parameter array has been triggered. If this mechanism is used, it must be the only array used in the DIR or block. The default data handling is determined by the last ZI or ZR card before the ZN card.

Warning: A dummy ZX card must immediately follow the ZM card if any ZP cards appear in the array specifications.

ZN Card Format

Field Number	Card Columns	Field Entry	Field Description
1	1-4	ZN	Z-Card Type Designator

The ZN card format is used to terminate lists from ZA and ZM card formats. It does not have any other function.

Appendix B

References

REFERENCES

1. The BDM Corporation; "Draft ETC^3 Computer Model User's Guide;" BDM/W-78-211-TR; April 10, 1978.

2. The BDM Corporation; "The System Support Translator (SST) User's Guide;" BDM/W-76-201-TR; December 15, 1976.

3. The BDM Corporation; "Nuclear Effects Routines for the ETC^3 Code;" Unpublished.

4. Defense Nuclear Agency; "Capabilities of Nuclear Weapons;" Unpublished.

5. Harry Diamond Laboratories; Nuclear Weapons Environments for Vulnerability Assessment to Support Tactical Nuclear Warfare Analyses; Unpublished.

6. Glasstone, Samuel, and Philip J. Dolan. THE EFFECTS OF NUCLEAR WEAPONS, 3rd Edition: 1977.

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

AFSOUTH
ATTN: U.S. Documents Officer

Armed Forces Staff College
ATTN: Reference & Technical Services Branch

Assistant Secretary of Defense
Comm., Cmd., Cont. & Intell.
ATTN: Assistant for NATO C3
ATTN: Strat. & Theater C&C System
ATTN: Surveillance & Warning Systems

Assistant Secretary of Defense
Program Analysis & Evaluation
ATTN: General Purpose Programs

Assistant Secretary of Defense
Manpower Reserve Affairs & Logistics
ATTN: Dep. Asst. Secy., Program Development

Assistant to the Secretary of Defense
Atomic Energy
ATTN: B. Adams

Command & Control Technical Center
ATTN: C-610
ATTN: C-650
ATTN: C-330

Commander-in-Chief
U.S. European Command
ATTN: ECJ6-0

Defense Advanced Rsch. Proj. Agency
ATTN: TIO

Defense Communications Agency
ATTN: Code 101B
ATTN: Code 205
ATTN: Code 410
ATTN: Code 510
ATTN: WWMCCS Sys. Engr.
ATTN: Code 520
ATTN: Code 530

Defense Communications Engineer Center
ATTN: Code R720
ATTN: Code R123
ATTN: Code R400

Defense Documentation Center
12 cy ATTN: DD

Defense Intelligence Agency
ATTN: DT-1C1
ATTN: RDS-4C
ATTN: DT-4B

Field Command
Defense Nuclear Agency
ATTN: FCPR

Field Command
Defense Nuclear Agency
Livermore Division
ATTN: FCPRL

DEPARTMENT OF DEFENSE (Continued)

Defense Nuclear Agency
ATTN: STVL
ATTN: DDST
ATTN: STSP
ATTN: OASO
ATTN: STNA
ATTN: STRA
ATTN: STSS
ATTN: VLIS
4 cy ATTN: TITL

Interservice Nuclear Weapons School
ATTN: TTV

Joint Chiefs of Staff
ATTN: J-3
ATTN: J-3, WWMCCS Evaluation Office
ATTN: J-3, Nuclear Contingency Branch
ATTN: J-3, NMCC
ATTN: J-5, Nuclear Division
ATTN: J-37
ATTN: SAGA

Joint Strat. Tgt. Planning Staff
ATTN: JPST
ATTN: JLTW-2
ATTN: NRI-STINFO Library
ATTN: JSAS

Joint Tactical Communications Office
ATTN: TT-E-SS

National Communications System
Office of the Manager
ATTN: NCS-TS

National Defense University
ATTN: NWCLB-CR

National Security Agency
ATTN: TDL
ATTN: R-52, O. Van Gunten

Director
Net Assessment
ATTN: Military Assistants

U.S. Forces, Korea
ATTN: DJ-AM-SM
ATTN: CJ-P-G

U.S. National Military Representative
SHAPE
10 cy ATTN: U.S. Documents Officer

Undersecretary of Defense for Rsch. & Engrg.
ATTN: Strategic & Space Systems (OS)
ATTN: Tactical Warfare Programs

DEPARTMENT OF THE ARMY

Asst. Chief of Staff for Intelligence
Department of the Army
ATTN: DAMI-FI

DEPARTMENT OF THE ARMY (Continued)

Atmospheric Sciences Laboratory
U.S. Army Electronics R & D Command
ATTN: DELAS-AS

Chief of Engineers
Department of the Army
ATTN: DAEN-MCE-D

Deputy Chief of Staff for Ops. & Plans
Department of the Army
ATTN: DAMO-ODW

Deputy Chief of Staff for Rsch. Dev. & Acq.
Department of the Army
ATTN: Advisor for RDA Analysis (M. Gale)
ATTN: DAMA-CSS-N

Harry Diamond Laboratories
Department of the Army
ATTN: DELHD-N-P
2 cy ATTN: DELHD-N-EM
ATTN: DELHD-N-RBA
ATTN: DELHD-N-CO
ATTN: DELHD-I-TL

Multi Service Communications Systems
Department of the Army
ATTN: DRCPM-MSCS-APB, M. Francis

U.S. Army Ballistic Research Laboratories
ATTN: TBL
ATTN: DRDAR-VL
ATTN: CAL

U.S. Army Comb. Arms Combat Dev. Acty.
ATTN: ATCA-CFT
ATTN: ATCA-CA
ATTN: ATCA-CO

U.S. Army Command & General Staff College
ATTN: ATSW-TA-D
ATTN: Library

U.S. Army Communications Sys. Agency
ATTN: CCM-AD-LB

U.S. Army Computer Systems Command
ATTN: Technical Library

U.S. Army Concepts Analysis Agency
ATTN: Code 605/606

U.S. Army Electronics Rsch. & Dev. Command
ATTN: DRCPM-TDS-SD
ATTN: DRCPM-ATC

Commander-in-Chief
U.S. Army Europe and Seventh Army
ATTN: ODCSE-E, AEAGE-PI
ATTN: AEAGB
ATTN: AEAGC-O

U.S. Army Materiel Dev. & Readiness Command
ATTN: DRCDE-D

U.S. Army Materiel Sys. Analysis Activity
ATTN: DRXSY-DS
ATTN: DRXSY-S

DEPARTMENT OF THE ARMY (Continued)

U.S. Army Missile R & D Command
ATTN: DRSMI-YDR

U.S. Army Mobility Equip. R & D Command
ATTN: DRDME-WC

U.S. Army Nuclear & Chemical Agency
ATTN: Library for MONA-SAL
ATTN: Library

U.S. Army Satellite Comm. Agency
ATTN: DRCPM-SC-11

U.S. Army Test and Evaluation Command
ATTN: DRSTE-EL
ATTN: Technical Library
ATTN: DRSTE-NB

U.S. Army TRADOC Systems Analysis Activity
ATTN: ATAA-TAC

U.S. Army Training and Doctrine Command
ATTN: ATORI-OP-SD
ATTN: Technical Library

U.S. Army War College
ATTN: Library

V Corps
Department of the Army
ATTN: AETVGB
ATTN: AETVCE
ATTN: AETVGC

VII Corps
Department of the Army
ATTN: AETSGB-O
ATTN: AETSCE
ATTN: AETSGC

DEPARTMENT OF THE NAVY

Naval Ocean Systems Center
ATTN: Research Library

Naval Surface Weapons Center
ATTN: Code F30
ATTN: Code F32

Naval Telecommunications Command
ATTN: Deputy Director, Systems
ATTN: Deputy Director, Plans

Office of the Chief of Naval Operations
ATTN: OP 94
ATTN: OP 604C
ATTN: OP 96

Commander-in-Chief
U.S. Naval Forces, Europe
ATTN: N3262, Nuclear Surety Officer

Commander-in-Chief
U.S. Atlantic Fleet
Department of the Navy
ATTN: Code J54
ATTN: Code J-611A

DEPARTMENT OF THE AIR FORCE

Aerospace Defense Command
Department of the Air Force
ATTN: ADCOM/INA

Air Force Geophysics Laboratory
ATTN: SULL

Air Force Weapons Laboratory, AFSC
ATTN: SUL
ATTN: DYC
ATTN: EL

Air University Library
Department of the Air Force
ATTN: AUL-LSE-70-250

Assistant Chief of Staff
Intelligence
Department of the Air Force
ATTN: INA

Assistant Chief of Staff
Studies & Analyses
Department of the Air Force
ATTN: AF/SASC

Deputy Chief of Staff
Operations Plans and Readiness
Department of the Air Force
ATTN: AFXOXFM
ATTN: AFXOK

Deputy Chief of Staff
Research, Development, & Acq.
Department of the Air Force
ATTN: AFRDQSM

Foreign Technology Division
Air Force Systems Command
ATTN: NIIS Library

Strategic Air Command
Department of the Air Force
ATTN: XPFS

Tactical Air Command
Department of the Air Force
ATTN: DRA

Commander-in-Chief
U.S. Air Forces in Europe
ATTN: XPXX
ATTN: DOC

U.S. Research & Development Coordinator
Department of the Air Force
10 cy ATTN: USRADCO

OTHER GOVERNMENT AGENCY

Central Intelligence Agency
ATTN: OSR/SEC
ATTN: OSR/SF, R. Virgo
ATTN: OSI/LSD, R. Hart

DEPARTMENT OF DEFENSE CONTRACTORS

BDM Corporation
ATTN: Corporate Library
ATTN: W. Sweeney
ATTN: L. Jacobs
ATTN: P. Bogue
ATTN: R. Labash
ATTN: J. Sanders

Boeing Company
ATTN: R. Scheppe

Computer Sciences Corporation
ATTN: H. Blank
ATTN: Library

E-Systems, Inc.
ECI Division
ATTN: J. Wilkes

ESL, Inc.
ATTN: Library
ATTN: J. Marshall

General Electric Company-TEMPO
ATTN: DASIAC

General Electric Company-TEMPO
ATTN: DASIAC

GTE Sylvania, Inc.
ATTN: M. Cross

Institute for Defense Analyses
ATTN: Classified Library

Kaman Sciences Corporation
ATTN: W. Long

Mission Research Corporation
ATTN: S. Gutsche

R & D Associates
ATTN: R. Schaefer
ATTN: C. MacDonald
ATTN: R. Poll

R & D Associates
ATTN: R. Latter

RCA Corporation
ATTN: E. Van Keuren

SRI International
ATTN: C. Shoens
ATTN: W. Jaye

TRW Defense & Space Systems Group
ATTN: R. Webb

TRW Defense & Space Systems Group
ATTN: J. Dyche

AD-A077 617 BDM CORP MCLEAN VA F/G 17/2
EUROPEAN THEATER COMMAND, CONTROL AND COMMUNICATIONS (ETC3) MOD--ETC(U)
NOV 78 P BOGUE, R LABASH, J SANDERS DNA001-78-C-0077
UNCLASSIFIED BDM/W-78-718-TR DNA-4351T-2B NL

4 OF 4
AD A077617

SUPPLEMENTARY INFORMATION

END
DATE FILMED
10-80
DTIC

SUPPLEMENTARY

INFORMATION

AD-A077617

ERRATA SHEET

for

DNA 4351T-2B dated 30 November 1979

Change the contract number on the front cover and in block 8 of the DD Form 1473 from DNA 001-78-C-0070 to DNA 001-78-C-[redacted].
0077